



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS



### BASE DE DATOS

PROFESOR:

Ing. Yadira Franco R

PERÍODO ACADÉMICO:

2025 A

### TAREA

TÍTULO:

TIPOS DE JOIN



Estudiante

Ayol Guanoluisa Nayely del Rocio.

Pérez Palacios Alessia de los Ángeles.

2025-A

# TRABAJO GRUPAL MAXIMO 2 INTEGRANTES

## TIPOS DE JOIN

El JOIN se usa para combinar filas de dos o más tablas basadas en una condición de relación entre ellas, generalmente usando claves foráneas.

### 2. Tipos de JOIN:

1. **INNER JOIN**: Este tipo de JOIN retorna solo las filas que tienen coincidencias en ambas tablas.
2. **LEFT JOIN (o LEFT OUTER JOIN)**: Retorna todas las filas de la tabla de la izquierda y las filas coincidentes de la tabla de la derecha. Si no hay coincidencias, devuelve NULL para la tabla de la derecha.
3. **RIGHT JOIN (o RIGHT OUTER JOIN)**: Similar al LEFT JOIN, pero devuelve todas las filas de la tabla de la derecha.
4. **FULL JOIN (o FULL OUTER JOIN)**: Retorna todas las filas cuando hay una coincidencia en una de las tablas, o ambas.

### ADICIONAL:

Agrega registros de prueba a las tablas correspondientes para garantizar que los resultados de cada tipo de JOIN sean representativos y completos. Los registros deben ser consistentes con los datos que se requieren para cada consulta y deben permitir visualizar claramente las diferencias en los resultados de cada JOIN.

## INSTRUCCIONES

1. **Sube un archivo al git hub** que contenga todos los elementos solicitados.
2. **Incluye los scripts SQL** correspondientes a cada tipo de JOIN (INNER, LEFT, RIGHT, FULL OUTER JOIN) con **comentarios detallados** explicando cada consulta.
3. **Adjunta las capturas de pantalla** de los resultados obtenidos para cada consulta, mostrando los datos de la ejecución de las consultas.

# CLÍNICA

## INNER JOIN

1. Relacionar **pacientes** con **consultas** cuando ambos están registrados.

```

119  -- CONSULTAS
120  -- INNER JOIN
121  -- 1. Relacionar pacientes con consultas cuando ambos están registrados.
122  -- Se usa inner join para combinar los registros de dos o mas tablas(con una condic:
123  -- En este caso la relacion es encontrar pacientes con consultas registradas
124  • select p.pacientesID, p.nombre as nombre_paciemte, p.apellido as apellido_paciente,
125  c.consultaID, c.fecha as fecha_consulta from pacientes p
126  inner join consultas c on p.pacientesID = c.pacienteID;
127

```

Result Grid

  Filter Rows:

Export:

 Wrap Cell Content:

	pacientesID	nombre_paciente	apellido_paciente	consultaID	fecha_consulta
▶	101	Laura	García	401	2025-05-10
	102	Carlos	Martínez	402	2025-05-11
	103	Ana	López	403	2025-05-12
	104	Javier	Sánchez	404	2025-05-13
	105	Marta	Fernández	405	2025-05-14
	106	David	Pérez	406	2025-05-15
	107	Sofía	Gómez	407	2025-05-16
	108	Daniel	Ruiz	408	2025-05-17
	109	Elena	Hernández	409	2025-05-18
	110	Pablo	Díaz	410	2025-05-19

## LEFT JOIN

1. Mostrar todos los **pacientes**, incluyendo aquellos que aún no tienen **consultas** agendadas.

```

152  -- LEFT JOIN
153  -- 1. Mostrar todos los pacientes, incluyendo aquellos q
154  -- Sirve para combinar los datos de las dos tablas mantenie
155  -- incluso cuando no hay coincidencias en la tabla derecha
156  • select p.pacientesID, p.nombre as nombre_paciente, p.apell:
157  c.consultaID, c.fecha as fecha_consulta
158  from pacientes p -- Esta es la tabla izquierda (primera ta
159  left join consultas c -- Esta tabla derecha solo contiene c
160  on p.pacientesID = c.pacienteID;

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	pacientesID	nombre_paciente	apellido_paciente	consultaID	fecha_consulta
▶	101	Laura	García	401	2025-05-10
	102	Carlos	Martínez	402	2025-05-11
	103	Ana	López	403	2025-05-12
	104	Javier	Sánchez	404	2025-05-13
	105	Marta	Fernández	405	2025-05-14
	106	David	Pérez	406	2025-05-15
	107	Sofía	Gómez	407	2025-05-16
	108	Daniel	Ruiz	408	2025-05-17
	109	Elena	Hernández	409	2025-05-18
	110	Pablo	Díaz	410	2025-05-19
	111	Alejandro	Martín	NULL	NULL
	112	Sara	González	NULL	NULL
	113	Jorge	Soler	NULL	NULL

2. Relacionar **consultas** con **doctores** asignados.

```
18 -- 2. Relacionar consultas con doctores asignados.
19 -- La relacion que comparten las tablas consultas y doctores es:
20 -- "los doctores asigandos a cada consulta"
21 • select c.consultaID, c.fecha as fecha_consulta,
22     d.doctorID, d.nombre as nombre_doctor,
23     d.apellido as apellido_doctor, d.especialidad as especialidad_doctor
24 from doctores d inner join consultas c on d.doctorID = c.doctorID;
```

consultaID	fecha_consulta	doctorID	nombre_doctor	apellido_doctor	especialidad_doctor
401	2025-05-10	201	Juan	Rodríguez	Cardiología
402	2025-05-11	202	María	Gutiérrez	Pediatría
403	2025-05-12	203	Luis	Moreno	Traumatología
404	2025-05-13	204	Carmen	Jiménez	Dermatología
405	2025-05-14	205	Pedro	Navarro	Neurología
406	2025-05-15	206	Teresa	Romero	Ginecología
407	2025-05-16	207	Alberto	Torres	Oftalmología
408	2025-05-17	208	Isabel	Serrano	Endocrinología
409	2025-05-18	209	Francisco	Molina	Psiquiatría
410	2025-05-19	210	Rosa	Ortega	Medicina General

2. Mostrar todas las **consultas**, incluyendo las que no tienen **doctor** asignado.

```
164 -- 2. Mostrar todas las consultas, incluyendo las que no
165 -- Si una consulta no tiene un doctor asignado se mostraran
166 • select c.consultaID, c.fecha as fecha_consulta,
167     d.nombre as nombre_doctor, d.apellido as apellido_doctor, d
168 from consultas c -- Tabla principal con todas las consultas
169 left join doctores d -- Tabla secundaria con las coincidenc
170 on c.doctorID =d.doctorID;
```

consultaID	fecha_consulta	nombre_doctor	apellido_doctor	especialidad_doctor
401	2025-05-10	Juan	Rodríguez	Cardiología
402	2025-05-11	María	Gutiérrez	Pediatría
403	2025-05-12	Luis	Moreno	Traumatología
404	2025-05-13	Carmen	Jiménez	Dermatología
405	2025-05-14	Pedro	Navarro	Neurología
406	2025-05-15	Teresa	Romero	Ginecología
407	2025-05-16	Alberto	Torres	Oftalmología
408	2025-05-17	Isabel	Serrano	Endocrinología
409	2025-05-18	Francisco	Molina	Psiquiatría
410	2025-05-19	Rosa	Ortega	Medicina General
411	2025-05-12	NULL	NULL	NULL
412	2025-05-15	NULL	NULL	NULL
413	2025-05-19	NULL	NULL	NULL

### 3. Relacionar **medicamentos** recetados con los **pacientes** que los reciben.

```
-- 3. Relacionar medicamentos recetados con los pacientes que los reciben.
-- Se relacionan cuatro tablas (medicamentos, recetas, pacientes y doctores)
select r.recetaID, r.fecha as fecha_receta, p.pacientesID, p.nombre as nombre_paciente,
p.apellido as apellido_paciente, m.medicamentoID,
m.nombre as nombre_medimento, d.nombre as nombre_doctor,
d.apellido as apellido_doctor from recetas r
-- Se relaciona una receta con el paciente al que le corresponde
inner join pacientes p on r.pacienteID = p.pacientesID
-- Relacion entre la receta y el medicamento que se ha prescrito
inner join medicamentos m on r.medicamentoID = m.medicamentoID
-- La receta se conecta con el doctor que la ha emitido
inner join doctores d on r.doctorID = d.doctorID;
```

recetaID	fecha_receta	pacientesID	nombre_paciente	apellido_paciente	medicamentoID	nombre_medimento	nombre_doctor	apellido_doctor
501	2025-05-10	101	Laura	García	301	Ibuprofeno	Juan	Rodríguez
502	2025-05-10	101	Laura	García	302	Paracetamol	Juan	Rodríguez
503	2025-05-11	102	Carlos	Martínez	303	Amoxicilina	María	Gutiérrez
504	2025-05-12	103	Ana	López	304	Omeprazol	Luis	Moreno
505	2025-05-13	104	Javier	Sánchez	305	Loratadina	Carmen	Jiménez
506	2025-05-14	105	Marta	Fernández	306	Simvastatina	Pedro	Navarro
507	2025-05-15	106	David	Pérez	307	Metformina	Teresa	Romero
508	2025-05-16	107	Sofía	Gómez	308	Salbutamol	Alberto	Torres
509	2025-05-17	108	Daniel	Ruiz	309	Diazepam	Isabel	Serrano
510	2025-05-18	109	Elena	Hernández	310	Atorvastatina	Francisco	Molina
511	2025-05-19	110	Pablo	Díaz	301	Ibuprofeno	Rosa	Ortega
512	2025-05-19	110	Pablo	Díaz	302	Paracetamol	Rosa	Ortega

### 3. Mostrar todos los **pacientes**, incluyendo aquellos que no reciben **medicamentos**.

```
173 -- 3. Mostrar todos los pacientes, incluyendo aquellos que no reciben medicamentos
174 -- Si un paciente no tiene una receta entonces el ID de la receta y el medicamento s
175 • select p.pacientesID, p.nombre as nombre_paciente, p.apellido as apellido_paciente,
176 r.recetaID, m.nombre as nombre_medimento
177 from pacientes p -- Tabla principal de la cual se mostraran todos los registros
178 -- Se conecta a los pacientes con las recetas, se mantienen todos los pacientes aunq
179 left join recetas r on p.pacientesID = r.pacienteID
180 -- Se conecta recetas con medicamentos, se aplica solo a los pacientes que tienen re
181 left join medicamentos m on r.medicamentoID = m.medicamentoID;
```

pacientesID	nombre_paciente	apellido_paciente	recetaID	nombre_medimento
101	Laura	García	501	Ibuprofeno
101	Laura	García	502	Paracetamol
102	Carlos	Martínez	503	Amoxicilina
103	Ana	López	504	Omeprazol
104	Javier	Sánchez	505	Loratadina
105	Marta	Fernández	506	Simvastatina
106	David	Pérez	507	Metformina
107	Sofía	Gómez	508	Salbutamol
108	Daniel	Ruiz	509	Diazepam
109	Elena	Hernández	510	Atorvastatina
110	Pablo	Díaz	511	Ibuprofeno
110	Pablo	Díaz	512	Paracetamol
111	Alejandro	Martín	NULL	NULL
112	Sara	González	NULL	NULL
113	Jorge	Soler	NULL	NULL

## RIGHT JOIN

- Mostrar todas las **consultas**, incluso aquellas que no están asignadas a un **paciente**.

```

190 -- 1.  Mostrar todas las consultas, incluso aquellas que no están asignadas a
191 -- Se muestran todas las consultas medicas y las que no tienen un paciente asig
192 • select c.consultaID, c.fecha as fecha_consulta,
193 p.pacientesID, p.nombre as nombre_paciente, p.apellido as apellido_paciente
194 from pacientes p -- Tabla secundaria que contiene los datos del los pacientes
195 right join consultas c -- Tabla principal con todos los datos de las consultas
196 on p.pacientesID = c.pacienteID;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	consultaID	fecha_consulta	pacientesID	nombre_paciente	apellido_paciente
▶	401	2025-05-10	101	Laura	García
	402	2025-05-11	102	Carlos	Martínez
	403	2025-05-12	103	Ana	López
	404	2025-05-13	104	Javier	Sánchez
	405	2025-05-14	105	Marta	Fernández
	406	2025-05-15	106	David	Pérez
	407	2025-05-16	107	Sofía	Gómez
	408	2025-05-17	108	Daniel	Ruiz
	409	2025-05-18	109	Elena	Hernández
	410	2025-05-19	110	Pablo	Díaz
	411	2025-05-12	NULL	NULL	NULL
	412	2025-05-15	NULL	NULL	NULL
	413	2025-05-19	NULL	NULL	NULL

## FULL OUTER JOIN

- Mostrar todos los **pacientes** y **consultas**, incluidas las que no están relacionadas entre sí.

```

-- 1.  Mostrar todos los pacientes y consultas, incluidas las que no están relacionadas
-- Todos los pacientes con sus consultas si existen
select p.pacientesID, p.nombre as nombre_paciente, p.apellido as apellido_paciente,
c.consultaID, c.fecha as fecha_consulta, d.nombre as doctor
from pacientes p -- Tabla principal, todos los registros se mantienen
left join consultas c on p.pacientesID = c.pacienteID -- Une con consultas si existen
left join doctores d on c.doctorID = d.doctorID -- Une doctor si hay consultas
union -- Combina resultados eliminando duplicados
-- Consultas sin pacientes
select p.pacientesID, p.nombre as nombre_paciente, p.apellido as apellido_paciente,
c.consultaID, c.fecha as fecha_consulta, d.nombre as nombre_doctor
from pacientes p
-- Prioriza todas las consultas (incluso sin pacientes)
right join consultas c on p.pacientesID = c.pacienteID
-- Añade información del doctor para cada consulta
left join doctores d on c.doctorID = d.doctorID
where p.pacientesID is null;

```

Activar Windows  
Ve a Configuración pa

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	pacientesID	nombre_paciente	apellido_paciente	consultaID	fecha_consulta	doctor
	101	Laura	García	401	2025-05-10	Juan
	102	Carlos	Martínez	402	2025-05-11	María
	103	Ana	López	403	2025-05-12	Luis
	104	Javier	Sánchez	404	2025-05-13	Carmen
	105	Marta	Fernández	405	2025-05-14	Pedro
	106	David	Pérez	406	2025-05-15	Teresa
	107	Sofía	Gómez	407	2025-05-16	Alberto
	108	Daniel	Ruiz	408	2025-05-17	Isabel
	109	Elena	Hernández	409	2025-05-18	Francisco
	110	Pablo	Díaz	410	2025-05-19	Rosa
▶	111	Alejandro	Martín	NULL	NULL	NULL
	112	Sara	González	NULL	NULL	NULL
	113	Jorge	Soler	NULL	NULL	NULL
	NULL	NULL	NULL	411	2025-05-12	NULL
	NULL	NULL	NULL	412	2025-05-15	NULL
	NULL	NULL	NULL	413	2025-05-19	NULL



2. Mostrar todos los **doctores**, incluso aquellos que no tienen **consultas** asignadas.

```

201 -- 2. Mostrar todos los doctores, incluso aquellos que no tienen consultas asignadas.
202 -- Se mantienen todos los registros de la tabla doctores y se relacionan con la tabla co
203 -- Si no hay consultas asignadas a un doctor se muestran valores nulos
204 • select d.doctorID, d.nombre as nombre_doctor, d.apellido as apellido_doctor,
205 d.especialidad as especialidad_doctor,
206 c.consultaID, c.fecha as fecha_consulta
207 from consultas c -- Tabla secundaria
208 right join doctores d -- Tabla principal
209 on d.doctorID = c.doctorID;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

doctorID	nombre_doctor	apellido_doctor	especialidad_doctor	consultaID	fecha_consulta
201	Juan	Rodríguez	Cardiología	401	2025-05-10
202	María	Gutiérrez	Pediatría	402	2025-05-11
203	Luis	Moreno	Traumatología	403	2025-05-12
204	Carmen	Jiménez	Dermatología	404	2025-05-13
205	Pedro	Navarro	Neurología	405	2025-05-14
206	Teresa	Romero	Ginecología	406	2025-05-15
207	Alberto	Torres	Oftalmología	407	2025-05-16
208	Isabel	Serrano	Endocrinología	408	2025-05-17
209	Francisco	Molina	Psiquiatría	409	2025-05-18
210	Rosa	Ortega	Medicina General	410	2025-05-19
211	Gabriela	Méndez	Oncología	NULL	NULL
212	Ricardo	Fernández	Cardiología Pediátrica	NULL	NULL
213	Patricia	Silva	Medicina Interna	NULL	NULL

2. Mostrar todas las **consultas** y **doctores**, incluso las no relacionadas.

```

-- 2. Mostrar todas las consultas y doctores, incluso las no relacionadas.
-- Muestra todos los doctores y los relacionan con la tabla consultas
-- Si no tienen consultas los valores se llenaran con null
select d.doctorID, d.nombre as nombre_doctor, d.apellido as apellido_doctor,
d.especialidad, c.consultaID, c.fecha as fecha_consulta, p.nombre as nombre_paciente
from doctores d
left join consultas c on d.doctorID = c.doctorID
left join pacientes p on c.pacienteID = p.pacientesID
union
-- Muestra consultas que no tienen un doctor asignado
select null as doctorID, null as nombre_doctor,
null as apellido_doctor, null as especialidad,
c.consultaID, c.fecha as fecha_consulta,
p.nombre as nombre_paciente
from consultas c
left join pacientes p on c.pacienteID = p.pacientesID
where c.doctorID is null -- Se filtran los registros de consultas sin doctor

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

doctorID	nombre_doctor	apellido_doctor	especialidad	consultaID	fecha_consulta	nombre_paciente
201	Juan	Rodríguez	Cardiología	401	2025-05-10	Laura
202	María	Gutiérrez	Pediatría	402	2025-05-11	Carlos
203	Luis	Moreno	Traumatología	403	2025-05-12	Ana
204	Carmen	Jiménez	Dermatología	404	2025-05-13	Javier
205	Pedro	Navarro	Neurología	405	2025-05-14	Marta
206	Teresa	Romero	Ginecología	406	2025-05-15	David
207	Alberto	Torres	Oftalmología	407	2025-05-16	Sofía
208	Isabel	Serrano	Endocrinología	408	2025-05-17	Daniel
209	Francisco	Molina	Psiquiatría	409	2025-05-18	Elena
210	Rosa	Ortega	Medicina General	410	2025-05-19	Pablo
211	Gabriela	Méndez	Oncología	NULL	NULL	NULL
212	Ricardo	Fernández	Cardiología Pedi...	NULL	NULL	NULL
213	Patricia	Silva	Medicina Interna	NULL	NULL	NULL
NULL	NULL	NULL	NULL	411	2025-05-12	NULL
NULL	NULL	NULL	NULL	412	2025-05-15	NULL
NULL	NULL	NULL	NULL	413	2025-05-19	NULL

3. Mostrar todos los **medicamentos**, incluso los que no han sido recetados.

```

217 -- 3.  Mostrar todos los medicamentos, incluso los que no han sido recetados.
218 -- Se mantienen todos los registros de la tabla medicamentos y se relacionan co
219 -- Si los medicamentos no tienen una receta se muestran valores nulos
220 • select m.medimentoID, m.nombre as nombre_medimento,
221      r.recetaID, r.fecha as fecha_receta
222 from recetas r -- Tabla secundaria
223 right join medicamentos m -- Tabla principal
224 on m.medimentoID = r.medimentoID;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	medimentoID	nombre_medimento	recetaID	fecha_receta
	301	Ibuprofeno	511	2025-05-19
	302	Paracetamol	502	2025-05-10
	302	Paracetamol	512	2025-05-20
	303	Amoxicilina	503	2025-05-11
	304	Omeprazol	504	2025-05-12
	305	Loratadina	505	2025-05-13
	306	Simvastatina	506	2025-05-14
	307	Metformina	507	2025-05-15
	308	Salbutamol	508	2025-05-16
	309	Diazepam	509	2025-05-17
	310	Atorvastatina	510	2025-05-18
	311	Rosuvastatina	NULL	NULL
	312	Metoprolol	NULL	NULL
	313	Losartán	NULL	NULL

3. Mostrar todos los **pacientes** y **medicamentos**, incluso los no relacionados entre sí.

```

-- 3.  Mostrar todos los pacientes y medicamentos, incluso los no relacionados entre sí.
select p.pacientesID, p.nombre as nombre_pacientes, p.apellido as apellido_paciente,
m.medimentoID, m.nombre as medicamento
from pacientes p -- Tabla principal
-- Conecta a los pacientes con las recetas
left join recetas r on p.pacientesID = r.pacienteID
-- Conecta recetas con medicamentos
left join medicamentos m on r.medimentoID = m.medimentoID
union -- Combina resultados eliminando duplicados
select null as pacientesID, null as nombre_paciente,
null as apellido_paciente, m.medimentoID, m.nombre as medicamento
from medicamentos m -- Todos los medicamentos
-- filtra solo medicamentos sin registros en recetas
where not exists (select 1 from recetas r where r.medimentoID = m.medimentoID)

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	pacientesID	nombre_pacientes	apellido_paciente	medimentoID	medicamento
	101	Laura	García	301	Ibuprofeno
	101	Laura	García	302	Paracetamol
	102	Carlos	Martínez	303	Amoxicilina
	103	Ana	López	304	Omeprazol
	104	Javier	Sánchez	305	Loratadina
	105	Marta	Fernández	306	Simvastatina
	106	David	Pérez	307	Metformina
	107	Sofía	Gómez	308	Salbutamol
	108	Daniel	Ruiz	309	Diazepam
	109	Elena	Hernández	310	Atorvastatina
	110	Pablo	Díaz	301	Ibuprofeno
	110	Pablo	Díaz	302	Paracetamol
	111	Alejandro	Martín	NULL	NULL
	112	Sara	González	NULL	NULL
	113	Jorge	Soler	NULL	NULL
	NULL	NULL	NULL	311	Rosuvastatina
	NULL	NULL	NULL	312	Metoprolol
	NULL	NULL	NULL	313	Losartán



## SCRIPT

```
create database clinicas; -- Se crea la base de datos clinica
use clinicas;
```

```
-- Se crea la tabla pacientes
```

```
create table pacientes(
    pacientesID int primary key not null, -- Se define la clave primaria y demas campos de la tabla
    nombre varchar(50) not null,
    apellido varchar(50) not null,
    edad int not null,
    ciudad varchar(50) not null,
    direccion varchar(100) not null
);
```

```
-- Se crea la tabla doctores
```

```
create table doctores(
    doctorID int primary key not null, -- Se define la clave primaria y demas campos de la tabla
    nombre varchar(50) not null,
    apellido varchar(50) not null,
    especialidad varchar(50) not null
);
```

```
-- Se crea la tabla medicamentos
```

```
create table medicamentos(
    medicamentoID int primary key not null, -- Se define la clave primaria y demas campos de la tabla
    nombre varchar(50) not null,
    descripcion varchar(100) not null
);
```

-- Se crea la tabla consultas

```
create table consultas(  
    consultaID int primary key not null, -- Se define la clave primaria y demas campos de la tabla  
    pacienteID int, -- Campo que almacenará el ID del paciente  
    foreign key (pacienteID) references pacientes(pacientesID), -- Se establece una relacion con la tabla pacientes  
    doctorID int,  
    foreign key (doctorID) references doctores(doctorID),  
    fecha date not null  
);
```

-- Se crea la tabla recetas

```
create table recetas(  
    recetaID int primary key not null, -- Se define la clave primaria y demas campos de la tabla  
    pacienteID int, -- Campo que almacenará el ID del paciente  
    foreign key (pacienteID) references pacientes(pacientesID), -- Se establece una relacion con la tabla pacientes  
    medicamentoID int, -- Campo que almacenará el ID del medicamento  
    foreign key (medicamentoID) references medicamentos(medicamentoID), -- Se establece relación con la tabla medicamentos  
    doctorID int, -- Campo que almacenará el ID del doctor  
    foreign key (doctorID) references doctores(doctorID), -- Se establece una relacion con la tabla doctores  
    fecha date not null  
);
```

-- Se insertan datos en la tabla pacientes

```
insert into pacientes(pacientesID, nombre, apellido, edad, ciudad, direccion) values  
(101, 'Laura', 'García', 28, 'Madrid', 'Calle Alcalá 123'),  
(102, 'Carlos', 'Martínez', 35, 'Barcelona', 'Avenida Diagonal 456'),  
(103, 'Ana', 'López', 42, 'Valencia', 'Calle Colón 789'),  
(104, 'Javier', 'Sánchez', 50, 'Sevilla', 'Plaza Nueva 321'),  
(105, 'Marta', 'Fernández', 22, 'Málaga', 'Calle Larios 654'),  
(106, 'David', 'Pérez', 65, 'Zaragoza', 'Paseo Independencia 101'),  
(107, 'Sofía', 'Gómez', 38, 'Bilbao', 'Gran Vía 202'),  
(108, 'Daniel', 'Ruiz', 29, 'Murcia', 'Plaza Circular 303'),  
(109, 'Elena', 'Hernández', 45, 'Palma', 'Paseo Marítimo 404'),  
(110, 'Pablo', 'Díaz', 31, 'Alicante', 'Explanada España 505'),
```

```
(111, 'Alejandro', 'Martín', 29, 'Alicante', 'Avenida de Denia 210'),  
(112, 'Sara', 'González', 33, 'Alicante', 'Calle San Fernando 75'),  
(113, 'Jorge', 'Soler', 30, 'Alicante', 'Paseo de la Explanada 12');
```

-- Se insertan datos en la tabla doctores

```
insert into doctores(doctorID, nombre, apellido, especialidad) values
```

```
(201, 'Juan', 'Rodríguez', 'Cardiología'),  
(202, 'María', 'Gutiérrez', 'Pediatría'),  
(203, 'Luis', 'Moreno', 'Traumatología'),  
(204, 'Carmen', 'Jiménez', 'Dermatología'),  
(205, 'Pedro', 'Navarro', 'Neurología'),  
(206, 'Teresa', 'Romero', 'Ginecología'),  
(207, 'Alberto', 'Torres', 'Oftalmología'),  
(208, 'Isabel', 'Serrano', 'Endocrinología'),  
(209, 'Francisco', 'Molina', 'Psiquiatría'),  
(210, 'Rosa', 'Ortega', 'Medicina General'),  
(211, 'Gabriela', 'Méndez', 'Oncología'),  
(212, 'Ricardo', 'Fernández', 'Cardiología Pediátrica'),  
(213, 'Patricia', 'Silva', 'Medicina Interna');
```

-- Se insertan datos en la tabla medicamentos

```
insert into medicamentos(medicamentoID, nombre, descripcion) values
```

```
(301, 'Ibuprofeno', 'Antiinflamatorio no esteroideo'),  
(302, 'Paracetamol', 'Analgésico y antipirético'),  
(303, 'Amoxicilina', 'Antibiótico betalactámico'),  
(304, 'Omeprazol', 'Inhibidor de la bomba de protones'),  
(305, 'Loratadina', 'Antihistamínico para alergias'),  
(306, 'Simvastatina', 'Para reducir el colesterol'),  
(307, 'Metformina', 'Antidiabético oral'),  
(308, 'Salbutamol', 'Broncodilatador para el asma'),  
(309, 'Diazepam', 'Ansiolítico y relajante muscular'),  
(310, 'Atorvastatina', 'Hipolipemiente'),  
(311, 'Rosuvastatina', 'Hipolipemiente de última generación'),  
(312, 'Metoprolol', 'Bloqueador beta para hipertensión'),
```

```
(313, 'Losartán', 'Antihipertensivo antagonista de receptores de angiotensina II');
```

```
-- Se insertan datos en la tabla consultas
```

```
insert into consultas(consultaID, pacienteID, doctorID, fecha) values
```

```
(401, 101, 201, '2025-05-10'),
```

```
(402, 102, 202, '2025-05-11'),
```

```
(403, 103, 203, '2025-05-12'),
```

```
(404, 104, 204, '2025-05-13'),
```

```
(405, 105, 205, '2025-05-14'),
```

```
(406, 106, 206, '2025-05-15'),
```

```
(407, 107, 207, '2025-05-16'),
```

```
(408, 108, 208, '2025-05-17'),
```

```
(409, 109, 209, '2025-05-18'),
```

```
(410, 110, 210, '2025-05-19'),
```

```
(411, null, null, '2025-05-12'),
```

```
(412, null, null, '2025-05-15'),
```

```
(413, null, null, '2025-05-19');
```

```
-- Se insertan datos en la tabla recetas
```

```
insert into recetas(recetaID, pacienteID, medicamentoID, doctorID, fecha) values
```

```
(501, 101, 301, 201, '2025-05-10'),
```

```
(502, 101, 302, 201, '2025-05-10'),
```

```
(503, 102, 303, 202, '2025-05-11'),
```

```
(504, 103, 304, 203, '2025-05-12'),
```

```
(505, 104, 305, 204, '2025-05-13'),
```

```
(506, 105, 306, 205, '2025-05-14'),
```

```
(507, 106, 307, 206, '2025-05-15'),
```

```
(508, 107, 308, 207, '2025-05-16'),
```

```
(509, 108, 309, 208, '2025-05-17'),
```

```
(510, 109, 310, 209, '2025-05-18'),
```

```
(511, 110, 301, 210, '2025-05-19'),
```

```
(512, 110, 302, 210, '2025-05-20'),
```

```
(513, null, null, 211, '2025-05-21'),
```

```
(514, null, null, 212, '2025-05-22'),
```

(515, null, null, 210, '2025-05-23');

-- CONSULTAS

-- INNER JOIN

-- Se usa inner join para combinar los registros de dos o mas tablas(con una condicion de relacion entre ellas), en este caso la tabla pacientes y la tabla consultas

-- 1. Relacionar pacientes con consultas cuando ambos están registrados.

-- En este caso la relacion es encontrar pacientes con consultas registradas

```
select p.pacientesID, p.nombre as nombre_paciente, p.apellido as apellido_paciente,  
c.consultaID, c.fecha as fecha_consulta from pacientes p  
inner join consultas c on p.pacientesID = c.pacienteID;
```

-- 2. Relacionar consultas con doctores asignados.

-- La relacion que comparten las tablas consultas y doctores es:

-- "los doctores asignados a cada consulta"

```
select c.consultaID, c.fecha as fecha_consulta,  
d.doctorID, d.nombre as nombre_doctor,  
d.apellido as apellido_doctor, d.especialidad as especialidad_doctor  
from doctores d inner join consultas c on d.doctorID = c.doctorID;
```

-- 3. Relacionar medicamentos recetados con los pacientes que los reciben.

-- Se relacionan cuatro tablas (medicamentos, recetas, pacientes y doctores)

```
select r.recetaID, r.fecha as fecha_receta, p.pacientesID, p.nombre as nombre_paciente,  
p.apellido as apellido_paciente, m.medimentoID,  
m.nombre as nombre_medimento, d.nombre as nombre_doctor,  
d.apellido as apellido_doctor from recetas r
```

-- Se relaciona una receta con el paciente al que le corresponde

```
inner join pacientes p on r.pacienteID = p.pacientesID
```

-- Relacion entre la receta y el medicamento que se ha prescrito

```
inner join medicamentos m on r.medimentoID = m.medimentoID
```

-- La receta se conecta con el doctor que la ha emitido

```
inner join doctores d on r.doctorID = d.doctorID;
```

-- LEFT JOIN

-- Sirve para combinar los datos de las dos tablas manteniendo los registros de la tabla izquierda

-- incluso cuando no hay coincidencias en la tabla derecha

-- 1.                      Mostrar todos los pacientes, incluyendo aquellos que aún no tienen consultas agendadas.

```
select p.pacientesID, p.nombre as nombre_paciente, p.apellido as apellido_paciente,
```

```
c.consultaID, c.fecha as fecha_consulta
```

```
from pacientes p -- Esta es la tabla izquierda (primera tabla mencionada) de la cual todos los registros se mantienen
```

```
left join consultas c -- Esta tabla derecha solo contiene coincidencias
```

```
on p.pacientesID = c.pacienteID;
```

-- 2.                      Mostrar todas las consultas, incluyendo las que no tienen doctor asignado.

-- Si una consulta no tiene un doctor asignado se mostraran las columnas con los datos de los doctores llena valores nulos

```
select c.consultaID, c.fecha as fecha_consulta,
```

```
d.nombre as nombre_doctor, d.apellido as apellido_doctor, d.especialidad as especialidad_doctor
```

```
from consultas c -- Tabla principal con todas las consultas
```

```
left join doctores d -- Tabla secundaria con las coincidencias
```

```
on c.doctorID = d.doctorID;
```

-- 3.                      Mostrar todos los pacientes, incluyendo aquellos que no reciben medicamentos.

-- Si un paciente no tiene una receta entonces el ID de la receta y el medicamento se mostraran con valores nulos

```
select p.pacientesID, p.nombre as nombre_paciente, p.apellido as apellido_paciente,
```

```
r.recetaID, m.nombre as nombre_medimento
```

```
from pacientes p -- Tabla principal de la cual se mostraran todos los registros
```

-- Se conecta a los pacientes con las recetas, se mantienen todos los pacientes aunque no tengan recetas

```
left join recetas r on p.pacientesID = r.pacienteID
```

-- Se conecta recetas con medicamentos, se aplica solo a los pacientes que tienen recetas

```
left join medicamentos m on r.medimentoID = m.medimentoID;
```

-- RIGHT JOIN

-- Es una operacion de union entre tablas que mantiene todos los registros de la tabla principal (segunda tabla mencionada)

-- y solo muestra los registros de coincidencias de la tabla secundaria (primera tabla mencionada)



-- 1.                      Mostrar todas las consultas, incluso aquellas que no están asignadas a un paciente.  
-- Se muestran todas las consultas medicas y las que no tienen un paciente asignado se muestran con valores nulos  
select c.consultaID, c.fecha as fecha\_consulta,  
p.pacientesID, p.nombre as nombre\_paciente, p.apellido as apellido\_paciente  
from pacientes p -- Tabla secundaria que contiene los datos de los pacientes  
right join consultas c -- Tabla principal con todos los datos de las consultas  
on p.pacientesID = c.pacienteID;

-- 2.                      Mostrar todos los doctores, incluso aquellos que no tienen consultas asignadas.  
-- Se mantienen todos los registros de la tabla doctores y se relacionan con la tabla consultas  
-- Si no hay consultas asignadas a un doctor se muestran valores nulos  
select d.doctorID, d.nombre as nombre\_doctor, d.apellido as apellido\_doctor,  
d.especialidad as especialidad\_doctor,  
c.consultaID, c.fecha as fecha\_consulta  
from consultas c -- Tabla secundaria  
right join doctores d -- Tabla principal  
on d.doctorID = c.doctorID;

-- 3.                      Mostrar todos los medicamentos, incluso los que no han sido recetados.  
-- Se mantienen todos los registros de la tabla medicamentos y se relacionan con la tabla recetas  
-- Si los medicamentos no tienen una receta se muestran valores nulos  
select m.medimentoID, m.nombre as nombre\_medimento,  
r.recetaID, r.fecha as fecha\_receta  
from recetas r -- Tabla secundaria  
right join medicamentos m -- Tabla principal  
on m.medimentoID = r.medimentoID;

-- FULL OUTER JOIN  
-- Es una operacion de union que preserva todos los registros de ambas tablas  
-- MySQL no soporta directamente FULL OUTER JOIN, por lo que debemos simularlo con una combinación de LEFT JOIN y RIGHT JOIN.

-- 1.                      Mostrar todos los pacientes y consultas, incluidas las que no están relacionadas entre sí.

```

-- Todos los pacientes con sus consultas si existen
select p.pacientesID, p.nombre as nombre_paciente, p.apellido as apellido_paciente,
c.consultaID, c.fecha as fecha_consulta, d.nombre as doctor
from pacientes p -- Tabla principal, todos los registros se mantienen
left join consultas c on p.pacientesID = c.pacienteID -- Une con consultas si existen
left join doctores d on c.doctorID = d.doctorID -- Une doctor si hay consultas
union -- Combina resultados eliminando duplicados
-- Consultas sin pacientes
select p.pacientesID, p.nombre as nombre_paciente, p.apellido as apellido_paciente,
c.consultaID, c.fecha as fecha_consulta, d.nombre as nombre_doctor
from pacientes p
-- Prioriza todas las consultas (incluso sin pacientes)
right join consultas c on p.pacientesID = c.pacienteID
-- Añade información del doctor para cada consulta
left join doctores d on c.doctorID = d.doctorID
where p.pacientesID is null;

```

```

-- 2.                Mostrar todas las consultas y doctores, incluso las no relacionadas.
-- Muestra todos los doctores y los relacionan con la tabla consultas
-- Si no tienen consultas los valores se llenaran con null
select d.doctorID, d.nombre as nombre_doctor, d.apellido as apellido_doctor,
d.especialidad, c.consultaID, c.fecha as fecha_consulta, p.nombre as nombre_paciente
from doctores d
left join consultas c on d.doctorID = c.doctorID
left join pacientes p on c.pacienteID = p.pacientesID
union
-- Muestra consultas que no tienen un doctor asignado
select null as doctorID, null as nombre_doctor,
null as apellido_doctor, null as especialidad,
c.consultaID, c.fecha as fecha_consulta,
p.nombre as nombre_paciente
from consultas c
left join pacientes p on c.pacienteID = p.pacientesID

```

```
where c.doctorID is null; -- Se filtran los registros de consultas sin doctor
```

```
-- 3.                Mostrar todos los pacientes y medicamentos, incluso los no relacionados entre sí.
```

```
select p.pacientesID, p.nombre as nombre_pacientes, p.apellido as apellido_paciente,  
m.medicamentoID, m.nombre as medicamento
```

```
from pacientes p -- Tabla principal
```

```
-- Conecta a los pacientes con las recetas
```

```
left join recetas r on p.pacientesID = r.pacienteID
```

```
-- Conecta recetas con medicamentos
```

```
left join medicamentos m on r.medicamentoID = m.medicamentoID
```

```
union -- Combina resultados eliminando duplicados
```

```
select null as pacientesID, null as nombre_paciente,
```

```
null as apellido_paciente, m.medicamentoID, m.nombre as medicamento
```

```
from medicamentos m -- Todos los medicamentos
```

```
-- filtra solo medicamentos sin registros en recetas
```

```
where not exists (select 1 from recetas r where r.medicamentoID = m.medicamentoID)
```

# CURSOS ONLINE

## INNER JOIN

1. Relacionar estudiantes con cursos en los que están inscritos.

```

102 -- CONSULTAS
103 -- INNER JOIN
104 -- 1) Relacionar estudiantes con cursos en los que están inscritos.
105 • select e.nombreEstudiante, c.nombreCurso
106 from estudiantes e
107 inner join inscripciones i on e.estudianteID = i.estudianteID
108 inner join cursos c on i.cursoID = c.cursoID;
109

```

nombreEstudiante	nombreCurso
Isabela Torres	Culinaria Profesional
Carlos Molina	Psicología General
Luna Márquez	Desarrollo de Software
Tomás Jiménez	Psicología Forense
Camila Solís	Inteligencia Artificial
Martín Duarte	Repostería Creativa
Daniela Ávila	Diseño de Bases de Datos
Andrés Roldán	Neuropsicología Aplicada
Florencia Peña	Hacking Ético
Benjamín León	Didáctica Infantil

## LEFT JOIN

1. Mostrar todos los **estudiantes, incluso aquellos que no están inscritos en ningún**

```

119 -- 1) Mostrar todos los estudiantes, incluso aquellos que no
120 • select e.nombreEstudiante, c.nombreCurso
121 from estudiantes e
122 left join inscripciones i on e.estudianteID = i.estudianteID
123 left join cursos c on i.cursoID = c.cursoID;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

nombreEstudiante	nombreCurso
Isabela Torres	Culinaria Profesional
Carlos Molina	Psicología General
Luna Márquez	Desarrollo de Software
Tomás Jiménez	Psicología Forense
Camila Solís	Inteligencia Artificial
Martín Duarte	Repostería Creativa
Daniela Ávila	Diseño de Bases de Datos
Andrés Roldán	Neuropsicología Aplicada
Florencia Peña	Hacking Ético
Benjamín León	Didáctica Infantil

## 2. Relacionar cursos con sus instructores

```

109 -- 2) Relacionar cursos con sus instructores.
110 • select c.nombreCurso, i.nombre as nombreInstructor
111 from cursos c
112 inner join instructores i on c.instructorID = i.instructorID;

```

Result Grid |   Filter Rows:  | Export:  | Wrap Cell Content: 



nombreCurso	nombreInstructor
▶ Culinaria Profesional	Damián Ríos
Psicología General	Jack Navarro
Desarrollo de Software	Magnus Silva
Psicología Forense	Jaziel Cruz
Inteligencia Artificial	Aitana Vega
Repostería Creativa	Zoe Medina
Diseño de Bases de Datos	Thiago Pérez
Neuropsicología Aplicada	Alma Delgado
Hacking Ético	Iván Herrera
Didáctica Infantil	Noa Estrada

## 2. Mostrar todos los **cursos**, incluyendo aquellos sin **instructor** asignado.

```

124 -- 2) Mostrar todos los cursos, incluyendo aquellos sin inst
125 • select c.nombreCurso, i.nombre as nombreInstructor
126 from cursos c
127 left join instructores i on c.instructorID = i.instructorID;

```

Result Grid |   Filter Rows:  | Export:  | Wrap Cell Content: 



nombreCurso	nombreInstructor
▶ Culinaria Profesional	Damián Ríos
Psicología General	Jack Navarro
Desarrollo de Software	Magnus Silva
Psicología Forense	Jaziel Cruz
Inteligencia Artificial	Aitana Vega
Repostería Creativa	Zoe Medina
Diseño de Bases de Datos	Thiago Pérez
Neuropsicología Aplicada	Alma Delgado
Hacking Ético	Iván Herrera
Didáctica Infantil	Noa Estrada

### 3. Relacionar estudiantes con lecciones completadas

```

113 -- 3) Relacionar estudiantes con lecciones completadas.
114 • select e.nombreEstudiante, lc.leccionesCompletadas, c.nombreCurso
115 from leccionescompletadas lc
116 inner join estudiantes e on lc.estudianteID = e.estudianteID
117 inner join cursos c on lc.cursoID = c.cursoID;

```



Result Grid			
Filter Rows: <input type="text"/>			
Export: 			
Wrap Cell Content: 			
	nombreEstudiante	leccionesCompletadas	nombreCurso
▶	Isabela Torres	5	Culinaria Profesional
	Carlos Molina	3	Psicología General
	Luna Márquez	4	Desarrollo de Software
	Tomás Jiménez	6	Psicología Forense
	Camila Solís	2	Inteligencia Artificial
	Martín Duarte	7	Repostería Creativa
	Daniela Ávila	5	Diseño de Bases de Datos
	Andrés Roldán	8	Neuropsicología Aplicada
	Florencia Peña	6	Hacking Ético
	Benjamín León	9	Didáctica Infantil

### 3. Mostrar todos los **estudiantes**, incluso aquellos que no han completado ninguna **lección**.

```

128 -- 3) Mostrar todos los estudiantes, incluso aquellos que no han comp
129 • select e.nombreEstudiante, c.nombreCurso, lc.leccionesCompletadas
130 from estudiantes e
131 left join leccionescompletadas lc on e.estudianteID = lc.estudianteID
132 left join cursos c on lc.cursoID = c.cursoID;

```

Result Grid			
Filter Rows: <input type="text"/>			
Export: 			
Wrap Cell Content: 			
	nombreEstudiante	nombreCurso	leccionesCompletadas
▶	Isabela Torres	Culinaria Profesional	5
	Carlos Molina	Psicología General	3
	Luna Márquez	Desarrollo de Software	4
	Tomás Jiménez	Psicología Forense	6
	Camila Solís	Inteligencia Artificial	2
	Martín Duarte	Repostería Creativa	7
	Daniela Ávila	Diseño de Bases de Datos	5
	Andrés Roldán	Neuropsicología Aplicada	8
	Florencia Peña	Hacking Ético	6
	Benjamín León	Didáctica Infantil	9



## RIGHT JOIN

1. Mostrar todos los cursos, incluso aquellos que no tienen estudiantes inscritos.

```

134 -- Mostrar todos los cursos, incluso aquellos que no tienen es
135 • select e.nombreEstudiante, c.nombreCurso
136 from estudiantes e
137 right join inscripciones i on e.estudianteID = i.estudianteID
138 right join cursos c on i.cursoID = c.cursoID;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

nombreEstudiante	nombreCurso
Isabela Torres	Culinaria Profesional
Carlos Molina	Psicología General
Luna Márquez	Desarrollo de Software
Tomás Jiménez	Psicología Forense
Camila Solís	Inteligencia Artificial
Martín Duarte	Repostería Creativa
Daniela Ávila	Diseño de Bases de Datos
Andrés Roldán	Neuropsicología Aplicada
Florencia Peña	Hacking Ético
Benjamín León	Didáctica Infantil

## FULL OUTER JOIN

1. Mostrar todos los **estudiantes** y **cursos**, incluso los no relacionados entre sí.

```

149 -- 1) Mostrar todos los estudiantes y cursos, incluso los no r
150 -- Estudiantes con sus cursos (incluso si no están inscritos)
151 • select e.nombreEstudiante, c.nombreCurso
152 from estudiantes e
153 left join inscripciones i on e.estudianteID = i.estudianteID
154 left join cursos c on i.cursoID = c.cursoID
155 union
156 select e.nombreEstudiante, c.nombreCurso
157 from estudiantes e
158 right join inscripciones i on e.estudianteID = i.estudianteID
159 right join cursos c on i.cursoID = c.cursoID;

```


Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

nombreEstudiante	nombreCurso
Isabela Torres	Culinaria Profesional
Carlos Molina	Psicología General
Luna Márquez	Desarrollo de Software
Tomás Jiménez	Psicología Forense
Camila Solís	Inteligencia Artificial
Martín Duarte	Repostería Creativa
Daniela Ávila	Diseño de Bases de Datos
Andrés Roldán	Neuropsicología Aplicada
Florencia Peña	Hacking Ético
Benjamín León	Didáctica Infantil

2. Mostrar todos los instructores, incluso aquellos que no tienen cursos a su cargo.


```
139 -- 2) Mostrar todos los instructores, incluso aquellos que no
140 • select c.nombreCurso, i.nombre as nombreInstructor
141 from cursos c
142 right join instructores i on c.instructorID = i.instructorID;
```

Result Grid




Filter Rows:

Export:



Wrap Cell Content:





nombreEstudiante	nombreCurso
Isabela Torres	Culinaria Profesional
Carlos Molina	Psicología General
Luna Márquez	Desarrollo de Software
Tomás Jiménez	Psicología Forense
Camila Solís	Inteligencia Artificial
Martín Duarte	Repostería Creativa
Daniela Ávila	Diseño de Bases de Datos
Andrés Roldán	Neuropsicología Aplicada
Florencia Peña	Hacking Ético
Benjamín León	Didáctica Infantil

2. Mostrar todos los **cursos** y **instructores**, incluidos los no relacionados entre sí.

```
160 -- 2) Mostrar todos los cursos y instructores, incluidos los
161 • select c.nombreCurso, i.nombre as nombreInstructor
162 from cursos c
163 left join instructores i on c.instructorID = i.instructorID
164 union
165 select c.nombreCurso, i.nombre as nombreInstructor
166 from cursos c
167 right join instructores i on c.instructorID = i.instructorID;
```


Result Grid






Filter Rows:

Export:



Wrap Cell Content:



nombreCurso	nombreInstructor
Culinaria Profesional	Damián Ríos
Psicología General	Jack Navarro
Desarrollo de Software	Magnus Silva
Psicología Forense	Jaziel Cruz
Inteligencia Artificial	Aitana Vega
Repostería Creativa	Zoe Medina
Diseño de Bases de Datos	Thiago Pérez
Neuropsicología Aplicada	Alma Delgado
Hacking Ético	Iván Herrera
Didáctica Infantil	Noa Estrada

3. Mostrar todas las lecciones, incluyendo las que no han sido completadas por ningún estudiante.

```

143 -- 3) Mostrar todas las lecciones, incluyendo las que no han sido comp
144 • select e.nombreEstudiante, c.nombreCurso, lc.leccionesCompletadas
145 from estudiantes e
146 right join leccionescompletadas lc on e.estudianteID = lc.estudianteID
147 right join cursos c on lc.cursoID = c.cursoID;

```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	nombreEstudiante	nombreCurso	leccionesCompletadas
►	Isabela Torres	Culinaria Profesional	5
	Carlos Molina	Psicología General	3
	Luna Márquez	Desarrollo de Software	4
	Tomás Jiménez	Psicología Forense	6
	Camila Solís	Inteligencia Artificial	2
	Martín Duarte	Repostería Creativa	7
	Daniela Ávila	Diseño de Bases de Datos	5
	Andrés Roldán	Neuropsicología Aplicada	8
	Florencia Peña	Hacking Ético	6
	Benjamín León	Didáctica Infantil	9

3. Mostrar todos los **estudiantes** y **lecciones**, incluso las no relacionadas entre sí.

```

168 -- 3) Mostrar todos los estudiantes y lecciones, incluso las no relac
169 • select e.nombreEstudiante, c.nombreCurso, lc.leccionesCompletadas
170 from estudiantes e
171 left join leccionescompletadas lc on e.estudianteID = lc.estudianteID
172 left join cursos c on lc.cursoID = c.cursoID
173 union
174 select e.nombreEstudiante, c.nombreCurso, lc.leccionesCompletadas
175 from estudiantes e
176 right join leccionescompletadas lc on e.estudianteID = lc.estudianteID
177 right join cursos c on lc.cursoID = c.cursoID;

```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	nombreEstudiante	nombreCurso	leccionesCompletadas
►	Isabela Torres	Culinaria Profesional	5
	Carlos Molina	Psicología General	3
	Luna Márquez	Desarrollo de Software	4
	Tomás Jiménez	Psicología Forense	6
	Camila Solís	Inteligencia Artificial	2
	Martín Duarte	Repostería Creativa	7
	Daniela Ávila	Diseño de Bases de Datos	5
	Andrés Roldán	Neuropsicología Aplicada	8
	Florencia Peña	Hacking Ético	6
	Benjamín León	Didáctica Infantil	9

## SCRIPT

```
-- creamos nuestra base de datos llamada cursos
create database cursos;
use cursos;

-- Creamos la tabla instructores
create table instructores(
-- Definimos como tal la clave primaria y los demás campos
instructorID int primary key auto_increment not null,
nombre varchar(100) not null,
especialidad varchar(100) not null
);

-- Creamos la tabla estudiantes
create table estudiantes(
-- Definimos la clave primaria y los demás campos
estudianteID int primary key auto_increment not null,
nombreEstudiante varchar(100) not null,
fechaInscripcion date not null
);

-- Creamos la tabla cursos
create table cursos(
-- Definimos la clave primaria y los demás campos
cursoID int primary key auto_increment not null,
nombreCurso varchar(100) not null,
fechaInscripcion date not null,
instructorID int,
foreign key(instructorID) references instructores(instructorID)
);

-- Creamos la tabla inscripciones
```

```
create table inscripciones(
-- Definimos la clave primaria y los demás campos
inscripcionID int primary key auto_increment not null,
estudianteID int,
foreign key(estudianteID) references estudiantes(estudianteID),
cursoID int,
foreign key(cursoID) references cursos(cursoID)
);

-- Creamos la tabla leccionescompletadas
create table leccionescompletadas(
-- Definimos la clave primaria y los demás campos
leccionID int primary key auto_increment not null,
leccionesCompletadas int not null,
estudianteID int,
foreign key(estudianteID) references estudiantes(estudianteID),
cursoID int,
foreign key(cursoID) references cursos(cursoID)
);

-- Se inserta datos en la tabla instructores
insert into instructores (nombre,especialidad) values
('Damián Ríos', 'Culinaria'),
('Jack Navarro', 'Psicología'),
('Magnus Silva', 'Desarrollo de Software'),
('Jaziel Cruz', 'Psicología Forense'),
('Aitana Vega', 'Inteligencia Artificial'),
('Zoe Medina', 'Pastelería'),
('Thiago Pérez', 'Bases de Datos'),
('Alma Delgado', 'Neuropsicología'),
('Iván Herrera', 'Ciberseguridad'),
('Noa Estrada', 'Educación Infantil');

-- Se inserta datos en la tabla estudiantes
```

```
insert into estudiantes(nombreEstudiante,fechaInscripcion) values
```

```
('Isabela Torres', '2024-09-01'),  
( 'Carlos Molina', '2024-09-03'),  
( 'Luna Márquez', '2024-09-05'),  
( 'Tomás Jiménez', '2024-09-07'),  
( 'Camila Solís', '2024-09-08'),  
( 'Martín Duarte', '2024-09-10'),  
( 'Daniela Ávila', '2024-09-12'),  
( 'Andrés Roldán', '2024-09-15'),  
( 'Floencia Peña', '2024-09-17'),  
( 'Benjamín León', '2024-09-20');
```

```
-- Se inserta datos en la tabla cursos
```

```
insert into cursos(nombreCurso,fechaInscripcion,instructorId)values
```

```
('Culinaria Profesional', '2024-09-01', 1),  
( 'Psicología General', '2024-09-02', 2),  
( 'Desarrollo de Software', '2024-09-03', 3),  
( 'Psicología Forense', '2024-09-04', 4),  
( 'Inteligencia Artificial', '2024-09-05', 5),  
( 'Repostería Creativa', '2024-09-06', 6),  
( 'Diseño de Bases de Datos', '2024-09-07', 7),  
( 'Neuropsicología Aplicada', '2024-09-08', 8),  
( 'Hacking Ético', '2024-09-09', 9),  
( 'Didáctica Infantil', '2024-09-10', 10);
```

```
-- Se inserta datos en la tabla inscripciones
```

```
insert into inscripciones(estudianteID, cursoID) values
```

```
(1, 1), (2, 2), (3, 3), (4, 4), (5, 5),  
(6, 6), (7, 7), (8, 8), (9, 9), (10, 10);
```

```
-- Se inserta datos en la tabla leccionescompletadas
```

```
insert into leccionescompletadas(leccionesCompletadas, estudianteID, cursoID) values
```

```
(5, 1, 1), (3, 2, 2), (4, 3, 3), (6, 4, 4), (2, 5, 5),  
(7, 6, 6), (5, 7, 7), (8, 8, 8), (6, 9, 9), (9, 10, 10);
```



-- CONSULTAS

-- INNER JOIN

-- 1) Relacionar estudiantes con cursos en los que están inscritos.

```
select e.nombreEstudiante, c.nombreCurso
```

```
from estudiantes e
```

```
inner join inscripciones i on e.estudianteID = i.estudianteID
```

```
inner join cursos c on i.cursoID = c.cursoID;
```

-- 2) Relacionar cursos con sus instructores.

```
select c.nombreCurso, i.nombre as nombreInstructor
```

```
from cursos c
```

```
inner join instructores i on c.instructorID = i.instructorID;
```

-- 3) Relacionar estudiantes con lecciones completadas.

```
select e.nombreEstudiante, lc.leccionesCompletadas, c.nombreCurso
```

```
from leccionescompletadas lc
```

```
inner join estudiantes e on lc.estudianteID = e.estudianteID
```

```
inner join cursos c on lc.cursoID = c.cursoID;
```

-- LEFT JOIN

-- 1) Mostrar todos los estudiantes, incluso aquellos que no están inscritos en ningún curso.

```
select e.nombreEstudiante, c.nombreCurso
```

```
from estudiantes e
```

```
left join inscripciones i on e.estudianteID = i.estudianteID
```

```
left join cursos c on i.cursoID = c.cursoID;
```

-- 2) Mostrar todos los cursos, incluyendo aquellos sin instructor asignado.

```
select c.nombreCurso, i.nombre as nombreInstructor
```

```
from cursos c
```

```
left join instructores i on c.instructorID = i.instructorID;
```

-- 3) Mostrar todos los estudiantes, incluso aquellos que no han completado ninguna lección.

```
select e.nombreEstudiante, c.nombreCurso, lc.leccionesCompletadas
```

```
from estudiantes e
```

```
left join leccionescompletadas lc on e.estudianteID = lc.estudianteID
```

```
left join cursos c on lc.cursoID = c.cursoID;
```

-- RIGHT JOIN

```

-- 1) Mostrar todos los cursos, incluso aquellos que no tienen estudiantes inscritos.
select e.nombreEstudiante, c.nombreCurso
from estudiantes e
right join inscripciones i on e.estudianteID = i.estudianteID
right join cursos c on i.cursoID = c.cursoID;
-- 2) Mostrar todos los instructores, incluso aquellos que no tienen cursos a su cargo.
select c.nombreCurso, i.nombre as nombreInstructor
from cursos c
right join instructores i on c.instructorID = i.instructorID;
-- 3) Mostrar todas las lecciones, incluyendo las que no han sido completadas por ningún estudiante.
select e.nombreEstudiante, c.nombreCurso, lc.leccionesCompletadas
from estudiantes e
right join leccionescompletadas lc on e.estudianteID = lc.estudianteID
right join cursos c on lc.cursoID = c.cursoID;
-- FULL OUTER JOIN
-- 1) Mostrar todos los estudiantes y cursos, incluso los no relacionados entre sí.
-- Estudiantes con sus cursos (incluso si no están inscritos)
select e.nombreEstudiante, c.nombreCurso
from estudiantes e
left join inscripciones i on e.estudianteID = i.estudianteID
left join cursos c on i.cursoID = c.cursoID
union
select e.nombreEstudiante, c.nombreCurso
from estudiantes e
right join inscripciones i on e.estudianteID = i.estudianteID
right join cursos c on i.cursoID = c.cursoID;
-- 2) Mostrar todos los cursos y instructores, incluidos los no relacionados entre sí.
select c.nombreCurso, i.nombre as nombreInstructor
from cursos c
left join instructores i on c.instructorID = i.instructorID
union
select c.nombreCurso, i.nombre as nombreInstructor
from cursos c
right join instructores i on c.instructorID = i.instructorID;

```

-- 3) Mostrar todos los estudiantes y lecciones, incluso las no relacionadas entre sí.

```
select e.nombreEstudiante, c.nombreCurso, lc.leccionesCompletadas
```

```
from estudiantes e
```

```
left join leccionescompletadas lc on e.estudianteID = lc.estudianteID
```

```
left join cursos c on lc.cursoID = c.cursoID
```

```
union
```

```
select e.nombreEstudiante, c.nombreCurso, lc.leccionesCompletadas
```

```
from estudiantes e
```

```
right join leccionescompletadas lc on e.estudianteID = lc.estudianteID
```

```
right join cursos c on lc.cursoID = c.cursoID;
```

# REDES SOCIALES

## INNER JOIN

4. Relacionar **usuarios** con sus **fotos** publicadas.

```

78  -- CONSULTAS
79  -- INNER JOIN
80  -- 1. Relacionar usuarios con sus fotos publicadas.
81  • select u.usuarioID, u.nombre, u.apellido,
82      f.fotoID, f.descripcion, f.fecha, f.me_gusta
83  from usuarios u
84  -- Se usa inner join para mostrar solo coincidencias exactas entre ambas tabla
85  inner join fotos f on u.usuarioID = f.usuarioID;
86

```

usuarioID	nombre	apellido	fotoID	descripcion	fecha	me_gusta
1	María	Gómez	101	Atardecer en la playa	2023-05-15	45
1	María	Gómez	104	Viaje a la montaña	2023-07-05	91
1	María	Gómez	108	Libro favorito	2023-01-30	28
2	Carlos	López	102	Mi nuevo perro	2023-06-20	78
2	Carlos	López	109	Jardín botánico	2023-09-14	87
3	Ana	Martínez	103	Comida casera	2023-04-10	32
4	Javier	Sánchez	105	Concierto de verano	2023-08-12	120
5	Lucía	Fernández	106	Día de spa	2023-03-18	65
6	Pedro	García	107	Nueva bicideta	2023-02-22	43

## LEFT JOIN

4. Mostrar todos los **usuarios**, incluyendo aquellos que no han subido ninguna **foto**.

```

104  -- LEFT JOIN
105  -- 1. Mostrar todos los usuarios, incluyendo aquellos que no han
106  -- Se muestran todos los registros, incluso los que no se relaci
107  • select u.usuarioID, u.nombre, u.apellido,
108      f.fotoID, f.descripcion, f.fecha, f.me_gusta
109  from usuarios u -- Se mantienen todos los registros de esta tabl
110  -- Mantiene todos los usuarios aunque no tengan fotos
111  left join fotos f -- Esta tabla solo contiene coincidencias
112  on u.usuarioID = f.usuarioID;

```

usuarioID	nombre	apellido	fotoID	descripcion	fecha	me_gusta
1	María	Gómez	101	Atardecer en la playa	2023-05-15	45
1	María	Gómez	104	Viaje a la montaña	2023-07-05	91
1	María	Gómez	108	Libro favorito	2023-01-30	28
2	Carlos	López	102	Mi nuevo perro	2023-06-20	78
2	Carlos	López	109	Jardín botánico	2023-09-14	87
3	Ana	Martínez	103	Comida casera	2023-04-10	32
4	Javier	Sánchez	105	Concierto de verano	2023-08-12	120
5	Lucía	Fernández	106	Día de spa	2023-03-18	65
6	Pedro	García	107	Nueva bicideta	2023-02-22	43
7	Sofía	Ruiz	NULL	NULL	NULL	NULL
8	David	Hernández	NULL	NULL	NULL	NULL
9	Elena	Jiménez	NULL	NULL	NULL	NULL
10	Pablo	Díaz	NULL	NULL	NULL	NULL

Activar W

5. Relacionar **fotos** con los **comentarios** recibidos.

```

87 -- 2. Relacionar fotos con los comentarios recibidos.
88 • select f.fotoID, f.descripcion as Descripcion_foto, f.fecha as fecha_foto,
89 c.comentarioID, c.comentario, c.fecha as fecha_comentario
90 from fotos f
91 -- Solo se mostraran fotos que tienen comentarios y comentarios que tienen fo
92 inner join comentarios c on f.fotoID=c.fotoID;
93 -- Se relaciona las fotos con sus comentarios
94

```

Result Grid						
Filter Rows:						
Export:						
Wrap Cell Content:						
fotoID	Descripcion_foto	fecha_foto	comentarioID	comentario	fecha_comentario	
101	Atardecer en la playa	2023-05-15	1001	iQué bonito atardecer!	2023-05-16	
101	Atardecer en la playa	2023-05-15	1002	Me encantaría estar ahí	2023-05-17	
102	Mi nuevo perro	2023-06-20	1005	Tu perro es adorable	2023-06-21	
104	Viaje a la montaña	2023-07-05	1003	Las vistas son impresionantes	2023-07-06	
105	Concierto de verano	2023-08-12	1004	iQué envidia!	2023-08-13	
106	Día de spa	2023-03-18	1006	Se ve relajante	2023-03-19	
107	Nueva bicicleta	2023-02-22	1007	Buena elección de modelo	2023-02-23	
108	Libro favorito	2023-01-30	1008	Lo leí el mes pasado, es genial	2023-01-31	
109	Jardín botánico	2023-09-14	1009	Hermosas flores	2023-09-15	
110	Cena con amigos	2023-10-05	1010	Se ve delicioso todo	2023-10-06	

5. Mostrar todas las **fotos**, incluyendo las que no tienen **comentarios**.

```

114 -- 2. Mostrar todas las fotos, incluyendo las que no tienen comentarios.
115 • select f.fotoID, f.descripcion, f.fecha, f.me_gusta,
116 c.comentarioID, c.comentario, c.fecha as fecha_comentario
117 from fotos f -- Se mantienen todos los registros de esta tabla
118 -- Mantiene todas las fotos aunque no tengan comentarios
119 left join comentarios c -- Solo contiene coincidencias
120 on f.fotoID = c.fotoID;

```

Result Grid						
Filter Rows:						
Export:						
Wrap Cell Content:						
fotoID	descripcion	fecha	me_gusta	comentarioID	comentario	fecha_comentario
101	Atardecer en la playa	2023-05-15	45	1001	iQué bonito atardecer!	2023-05-16
101	Atardecer en la playa	2023-05-15	45	1002	Me encantaría estar ahí	2023-05-17
102	Mi nuevo perro	2023-06-20	78	1005	Tu perro es adorable	2023-06-21
103	Comida casera	2023-04-10	32	NULL	NULL	NULL
104	Viaje a la montaña	2023-07-05	91	1003	Las vistas son impresionantes	2023-07-06
105	Concierto de verano	2023-08-12	120	1004	iQué envidia!	2023-08-13
106	Día de spa	2023-03-18	65	1006	Se ve relajante	2023-03-19
107	Nueva bicicleta	2023-02-22	43	1007	Buena elección de modelo	2023-02-23
108	Libro favorito	2023-01-30	28	1008	Lo leí el mes pasado, es genial	2023-01-31
109	Jardín botánico	2023-09-14	87	1009	Hermosas flores	2023-09-15
110	Cena con amigos	2023-10-05	95	1010	Se ve delicioso todo	2023-10-06
111	Deporte al aire libre	2023-11-11	110	NULL	NULL	NULL
112	Museo de arte	2023-12-03	52	NULL	NULL	NULL

6. Relacionar **usuarios** con sus **comentarios** realizados

```

95  -- 3. Relacionar usuarios con sus comentarios realizados.
96  • select u.usuarioID, u.nombre, u.apellido,
97      c.comentarioID, c.comentario, c.fecha as fecha_comentario
98  from usuarios u
99  -- Solo se muestran los usuarios con comentarios
100  inner join comentarios c on u.usuarioID = c.usuarioID;
101  -- Relaciona usuarios con sus comentarios a traves del ID
102

```

usuarioID	nombre	apellido	comentarioID	comentario	fecha_comentario
1	María	Gómez	1003	Las vistas son impresionantes	2023-07-06
1	María	Gómez	1006	Se ve relajante	2023-03-19
2	Carlos	López	1001	iQué bonito atardecer!	2023-05-16
3	Ana	Martínez	1002	Me encantaría estar ahí	2023-05-17
3	Ana	Martínez	1004	iQué envidia!	2023-08-13
3	Ana	Martínez	1007	Buena elección de modelo	2023-02-23
4	Javier	Sánchez	1005	Tu perro es adorable	2023-06-21
5	Lucía	Fernández	1008	Lo leí el mes pasado, es genial	2023-01-31
6	Pedro	García	1009	Hermosas flores	2023-09-15
6	Pedro	García	1010	Se ve delicioso todo	2023-10-06

6. Mostrar todos los **usuarios**, incluyendo aquellos que no han realizado ningún **comentario**.

```

122  -- 3. Mostrar todos los usuarios, incluyendo aquellos que no han reali:
123  • select u.usuarioID, u.nombre, u.apellido,
124      c.comentarioID, c.comentario, c.fecha as fecha_comentario
125  from usuarios u -- Se mantienen todos los registros de esta tabla
126  -- Mantiene todos los usuarios aunque no tengan comentarios
127  left join comentarios c -- Solo contiene coincidencias
128  on u.usuarioID = c.usuarioID;

```

usuarioID	nombre	apellido	comentarioID	comentario	fecha_comentario
1	María	Gómez	1003	Las vistas son impresionantes	2023-07-06
1	María	Gómez	1006	Se ve relajante	2023-03-19
2	Carlos	López	1001	iQué bonito atardecer!	2023-05-16
3	Ana	Martínez	1002	Me encantaría estar ahí	2023-05-17
3	Ana	Martínez	1004	iQué envidia!	2023-08-13
3	Ana	Martínez	1007	Buena elección de modelo	2023-02-23
4	Javier	Sánchez	1005	Tu perro es adorable	2023-06-21
5	Lucía	Fernández	1008	Lo leí el mes pasado, es genial	2023-01-31
6	Pedro	García	1009	Hermosas flores	2023-09-15
6	Pedro	García	1010	Se ve delicioso todo	2023-10-06
7	Sofía	Ruiz	NULL	NULL	NULL
8	David	Hernández	NULL	NULL	NULL
9	Elena	Jiménez	NULL	NULL	NULL
10	Pablo	Díaz	NULL	NULL	NULL



## RIGHT JOIN

4. Mostrar todas las **fotos**, incluso las que no están asociadas a ningún **usuario**.

```

130 -- RIGHT JOIN
131 -- 1) Mostrar todas las fotos, incluso las que no están asociadas a ningún usuario
132 • select
133 u.usuarioID, u.nombre, u.apellido, -- Datos del usuario pueden salir nulos si no
134 f.fotoID, f.descripcion, f.fecha, f.me_gusta -- Datos de las fotos siempre se muestran
135 from usuarios u
136 -- Se usa right join para mantener todas las fotos aunque no tengan usuario
137 right join fotos f on u.usuarioID = f.usuarioID;

```

usuarioID	nombre	apellido	fotoID	descripcion	fecha	me_gusta
1	María	Gómez	101	Atardecer en la playa	2023-05-15	45
2	Carlos	López	102	Mi nuevo perro	2023-06-20	78
3	Ana	Martínez	103	Comida casera	2023-04-10	32
1	María	Gómez	104	Viaje a la montaña	2023-07-05	91
4	Javier	Sánchez	105	Concierto de verano	2023-08-12	120
5	Lucía	Fernández	106	Día de spa	2023-03-18	65
6	Pedro	García	107	Nueva bicicleta	2023-02-22	43
1	María	Gómez	108	Libro favorito	2023-01-30	28
2	Carlos	López	109	Jardín botánico	2023-09-14	87
NULL	NULL	NULL	110	Cena con amigos	2023-10-05	95
NULL	NULL	NULL	111	Deporte al aire libre	2023-11-11	110
NULL	NULL	NULL	112	Museo de arte	2023-12-03	52

## FULL OUTER JOIN

4. Mostrar todos los **usuarios** y **fotos**, incluidas las que no tienen relación entre sí.

```

156 -- 1) Mostrar todos los usuarios y fotos, incluidas las que no tienen relación
157 -- Usuarios que tienen o no tienen fotos
158 • select
159 u.usuarioID, u.nombre, u.apellido,
160 f.fotoID, f.descripcion, f.fecha, f.me_gusta
161 from usuarios u
162 left join fotos f on u.usuarioID = f.usuarioID;
163
164 -- Fotos que tienen o no tienen usuario, por si hay fotos sin autor
165 • select
166 u.usuarioID, u.nombre, u.apellido,
167 f.fotoID, f.descripcion, f.fecha, f.me_gusta
168 from usuarios u
169 right join fotos f on u.usuarioID = f.usuarioID;

```

usuarioID	nombre	apellido	fotoID	descripcion	fecha	me_gusta
1	María	Gómez	101	Atardecer en la playa	2023-05-15	45
2	Carlos	López	102	Mi nuevo perro	2023-06-20	78
3	Ana	Martínez	103	Comida casera	2023-04-10	32
1	María	Gómez	104	Viaje a la montaña	2023-07-05	91
4	Javier	Sánchez	105	Concierto de verano	2023-08-12	120
5	Lucía	Fernández	106	Día de spa	2023-03-18	65
6	Pedro	García	107	Nueva bicicleta	2023-02-22	43
1	María	Gómez	108	Libro favorito	2023-01-30	28
2	Carlos	López	109	Jardín botánico	2023-09-14	87
NULL	NULL	NULL	110	Cena con amigos	2023-10-05	95
NULL	NULL	NULL	111	Deporte al aire libre	2023-11-11	110
NULL	NULL	NULL	112	Museo de arte	2023-12-03	52

5. Mostrar todos los **comentarios**, incluso los que no están asociados a ninguna **foto**.

```
139 -- 2) Mostrar todos los comentarios, incluso los que no están asociados
140 • select
141     f.fotoID, f.descripcion, f.fecha, -- Datos de la foto, pueden salir n
142     c.comentarioID, c.comentario, c.fecha as fecha_comentario -- Datos d
143 from fotos f
144 -- right join mantiene todos los comentarios aunque no tengan foto válida
145 right join comentarios c on f.fotoID = c.fotoID;
```

Result Grid						
Filter Rows: <input type="text"/> Export:  Wrap Cell Content: <input type="checkbox"/>						
	usuarioID	nombre	apellido	fotoID	descripcion	fecha
▶	1	María	Gómez	101	Atardecer en la playa	2023-05-15
	2	Carlos	López	102	Mi nuevo perro	2023-06-20
	3	Ana	Martínez	103	Comida casera	2023-04-10
	1	María	Gómez	104	Viaje a la montaña	2023-07-05
	4	Javier	Sánchez	105	Concierto de verano	2023-08-12
	5	Lucía	Fernández	106	Día de spa	2023-03-18
	6	Pedro	García	107	Nueva bicicleta	2023-02-22
	1	María	Gómez	108	Libro favorito	2023-01-30
	2	Carlos	López	109	Jardín botánico	2023-09-14
	NULL	NULL	NULL	110	Cena con amigos	2023-10-05
	NULL	NULL	NULL	111	Deporte al aire libre	2023-11-11
	NULL	NULL	NULL	112	Museo de arte	2023-12-03

5. Mostrar todas las **fotos y comentarios**, incluidos los no relacionados entre sí.

```
171 -- 2) Mostrar todas las fotos y comentarios, incluidos los no relacionados entre sí
172 -- Fotos que pueden tener comentarios o no
173 • select
174     f.fotoID, f.descripcion, f.fecha, f.me_gusta,
175     c.comentarioID, c.comentario, c.fecha as fecha_comentario
176 from fotos f
177 left join comentarios c on f.fotoID = c.fotoID;
178
179 -- Comentarios que pueden no estar relacionados con ninguna foto
180 • select
181     f.fotoID, f.descripcion, f.fecha, f.me_gusta,
182     c.comentarioID, c.comentario, c.fecha as fecha_comentario
183 from fotos f
184 right join comentarios c on f.fotoID = c.fotoID;
```

Result Grid						
Filter Rows: <input type="text"/> Export:  Wrap Cell Content: <input type="checkbox"/>						
	fotoID	descripcion	fecha	me_gusta	comentarioID	comentario
▶	101	Atardecer en la playa	2023-05-15	45	1001	iQué bonito atardecer!
	101	Atardecer en la playa	2023-05-15	45	1002	Me encantaría estar ahí
	104	Viaje a la montaña	2023-07-05	91	1003	Las vistas son impresionantes
	105	Concierto de verano	2023-08-12	120	1004	iQué envidia!
	102	Mi nuevo perro	2023-06-20	78	1005	Tu perro es adorable
	106	Día de spa	2023-03-18	65	1006	Se ve relajante
	107	Nueva bicicleta	2023-02-22	43	1007	Buena elección de modelo
	108	Libro favorito	2023-01-30	28	1008	Lo leí el mes pasado, es genial
	109	Jardín botánico	2023-09-14	87	1009	Hermosas flores
	110	Cena con amigos	2023-10-05	95	1010	Se ve delicioso todo
	NULL	NULL	NULL	NULL	1011	Qué energía
	NULL	NULL	NULL	NULL	1012	Fui la semana pasada
	NULL	NULL	NULL	NULL	1013	Espectacular foto astronómica

6. Mostrar todos los **comentarios**, incluso los que no están asociados a ningún **usuario**.

```

147 -- 3) Mostrar todos los comentarios, incluso los que no están asociados a ningún
148 • select
149     u.usuarioID, u.nombre, u.apellido, -- Datos del usuario, pueden ser nulos
150     c.comentarioID, c.comentario, c.fecha as fecha_comentario -- Datos del comentario
151 from usuarios u
152 -- Se usa RIGHT JOIN para asegurar que todos los comentarios aparezcan aunque no
153 right join comentarios c on u.usuarioID = c.usuarioID;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	fotoID	descripcion	fecha	comentarioID	comentario	fecha_comentario
▶	101	Atardecer en la playa	2023-05-15	1001	iQué bonito atardecer!	2023-05-16
	101	Atardecer en la playa	2023-05-15	1002	Me encantaría estar ahí	2023-05-17
	104	Viaje a la montaña	2023-07-05	1003	Las vistas son impresionantes	2023-07-06
	105	Concierto de verano	2023-08-12	1004	iQué envidia!	2023-08-13
	102	Mi nuevo perro	2023-06-20	1005	Tu perro es adorable	2023-06-21
	106	Día de spa	2023-03-18	1006	Se ve relajante	2023-03-19
	107	Nueva bicicleta	2023-02-22	1007	Buena elección de modelo	2023-02-23
	108	Libro favorito	2023-01-30	1008	Lo leí el mes pasado, es genial	2023-01-31
	109	Jardín botánico	2023-09-14	1009	Hermosas flores	2023-09-15
	110	Cena con amigos	2023-10-05	1010	Se ve delicioso todo	2023-10-06
	NULL	NULL	NULL	1011	Qué energía	2023-11-12
	NULL	NULL	NULL	1012	Fui la semana pasada	2023-12-04
	NULL	NULL	NULL	1013	Espectacular foto astronómica	2023-04-26

6. Mostrar todos los **usuarios** y **comentarios**, incluidos los no relacionados entre sí.

```

187 -- 3) Mostrar todos los usuarios y comentarios, incluidos los no relacionados
188 -- Usuarios que pueden tener o no tener comentarios
189 • select
190     u.usuarioID, u.nombre, u.apellido,
191     c.comentarioID, c.comentario, c.fecha as fecha_comentario
192 from usuarios u
193 left join comentarios c on u.usuarioID = c.usuarioID;
194
195 -- Comentarios que pueden no tener usuario
196 • select
197     u.usuarioID, u.nombre, u.apellido,
198     c.comentarioID, c.comentario, c.fecha as fecha_comentario
199 from usuarios u
200 right join comentarios c on u.usuarioID = c.usuarioID;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	usuarioID	nombre	apellido	comentarioID	comentario	fecha_comentario
▶	1	María	Gómez	1003	Las vistas son impresionantes	2023-07-06
	1	María	Gómez	1006	Se ve relajante	2023-03-19
	2	Carlos	López	1001	iQué bonito atardecer!	2023-05-16
	3	Ana	Martínez	1002	Me encantaría estar ahí	2023-05-17
	3	Ana	Martínez	1004	iQué envidia!	2023-08-13
	3	Ana	Martínez	1007	Buena elección de modelo	2023-02-23
	4	Javier	Sánchez	1005	Tu perro es adorable	2023-06-21
	5	Lucía	Fernández	1008	Lo leí el mes pasado, es genial	2023-01-31
	6	Pedro	García	1009	Hermosas flores	2023-09-15
	6	Pedro	García	1010	Se ve delicioso todo	2023-10-06
	7	Sofía	Ruiz	NULL	NULL	NULL
	8	David	Hernández	NULL	NULL	NULL
	9	Elena	Jiménez	NULL	NULL	NULL
	10	Pablo	Díaz	NULL	NULL	NULL

## SCRIPT

-- Crear la base de datos

```
create database red_social;
```

```
use red_social;
```

-- Se crea la tabla usuarios

```
create table usuarios (
```

```
    usuarioID int primary key not null, -- Se define la clave primaria y demas campos de la tabla
```

```
    nombre varchar(50) not null,
```

```
    apellido varchar(50) not null,
```

```
    edad int not null,
```

```
    ciudad varchar(50) not null,
```

```
    seguidores int not null,
```

```
    fotos_publicadas int not null
```

```
);
```

-- Se crea la tabla fotos

```
create table fotos (
```

```
    fotoID int primary key not null, -- Se define la clave primaria y demas campos de la tabla
```

```
    usuarioID int, -- Campo que almacenara el ID del usuario
```

```
    foreign key (usuarioID) references usuarios(usuarioID), -- Se establece una relacion con la tabla usuarios
```

```
    descripcion varchar(255) not null,
```

```
    fecha date not null,
```

```
    me_gusta int not null
```

```
);
```

-- Se crea la tabla comentarios

```
create table comentarios (
```

```
    comentarioID int primary key not null, -- Se define la clave primaria y demas campos de la tabla
```

```
    fotoID int, -- Campo que almacenara el ID del de las fotos
```

```
    foreign key (fotoID) references fotos(fotoID), -- Se establece una relacion con la tabla fotos
```

```
    usuarioID int, -- Campo que almacenara el ID del usuario
```

```
foreign key (usuarioid) references usuarios(usuarioid), -- Se establece una relacion con la tabla usuarios
comentario text not null,
fecha date not null
);
```

```
insert into usuarios (usuarioid, nombre, apellido, edad, ciudad, seguidores, fotos_publicadas) values
(1, 'María', 'Gómez', 25, 'Madrid', 350, 12),
(2, 'Carlos', 'López', 30, 'Barcelona', 420, 8),
(3, 'Ana', 'Martínez', 22, 'Valencia', 180, 5),
(4, 'Javier', 'Sánchez', 28, 'Sevilla', 600, 15),
(5, 'Lucía', 'Fernández', 35, 'Bilbao', 290, 7),
(6, 'Pedro', 'García', 40, 'Málaga', 150, 3),
(7, 'Sofía', 'Ruiz', 27, 'Zaragoza', 380, 9),
(8, 'David', 'Hernández', 31, 'Murcia', 210, 6),
(9, 'Elena', 'Jiménez', 29, 'Palma', 470, 11),
(10, 'Pablo', 'Díaz', 33, 'Alicante', 520, 14);
```

```
insert into fotos (fotoID, usuarioid, descripcion, fecha, me_gusta) values
(101, 1, 'Atardecer en la playa', '2023-05-15', 45),
(102, 2, 'Mi nuevo perro', '2023-06-20', 78),
(103, 3, 'Comida casera', '2023-04-10', 32),
(104, 1, 'Viaje a la montaña', '2023-07-05', 91),
(105, 4, 'Concierto de verano', '2023-08-12', 120),
(106, 5, 'Día de spa', '2023-03-18', 65),
(107, 6, 'Nueva bicicleta', '2023-02-22', 43),
(108, 1, 'Libro favorito', '2023-01-30', 28),
(109, 2, 'Jardín botánico', '2023-09-14', 87),
(110, null, 'Cena con amigos', '2023-10-05', 95),
(111, null, 'Deporte al aire libre', '2023-11-11', 110),
(112, null, 'Museo de arte', '2023-12-03', 52);
```

```
insert into comentarios (comentarioid, fotoID, usuarioid, comentario, fecha) values
(1001, 101, 2, '¡Qué bonito atardecer!', '2023-05-16'),
(1002, 101, 3, 'Me encantaría estar ahí', '2023-05-17'),
```

```
(1003, 104, 1, 'Las vistas son impresionantes', '2023-07-06'),  
(1004, 105, 3, '¡Qué envidia!', '2023-08-13'),  
(1005, 102, 4, 'Tu perro es adorable', '2023-06-21'),  
(1006, 106, 1, 'Se ve relajante', '2023-03-19'),  
(1007, 107, 3, 'Buena elección de modelo', '2023-02-23'),  
(1008, 108, 5, 'Lo leí el mes pasado, es genial', '2023-01-31'),  
(1009, 109, 6, 'Hermosas flores', '2023-09-15'),  
(1010, 110, 6, 'Se ve delicioso todo', '2023-10-06'),  
(1011, null, null, 'Qué energía', '2023-11-12'),  
(1012, null, null, 'Fui la semana pasada', '2023-12-04'),  
(1013, null, null, 'Espectacular foto astronómica', '2023-04-26');
```

-- CONSULTAS

-- INNER JOIN

-- 1. Relacionar usuarios con sus fotos publicadas.

```
select u.usuarioID, u.nombre, u.apellido,  
f.fotoID, f.descripcion, f.fecha, f.me_gusta  
from usuarios u
```

-- Se usa inner join para mostrar solo coincidencias exactas entre ambas tablas

```
inner join fotos f on u.usuarioID = f.usuarioID;
```

-- 2. Relacionar fotos con los comentarios recibidos.

```
select f.fotoID, f.descripcion as Descripcion_foto, f.fecha as fecha_foto,  
c.comentarioID, c.comentario, c.fecha as fecha_comentario  
from fotos f
```

-- Solo se mostraran fotos que tienen comentarios y comentarios que tienen fotos válidas

```
inner join comentarios c on f.fotoID=c.fotoID;
```

-- Se relaciona las fotos con sus comentarios

-- 3. Relacionar usuarios con sus comentarios realizados.

```
select u.usuarioID, u.nombre, u.apellido,  
c.comentarioID, c.comentario, c.fecha as fecha_comentario  
from usuarios u
```

-- Solo se muestran los usuarios con comentarios

```
inner join comentarios c on u.usuarioID = c.usuarioID;  
-- Relaciona usuarios con sus comentarios a traves del ID
```

```
-- LEFT JOIN
```

```
-- 1. Mostrar todos los usuarios, incluyendo aquellos que no han subido ninguna foto.
```

```
-- Se muestran todos los registros, incluso los que no se relacionan
```

```
select u.usuarioID, u.nombre, u.apellido,
```

```
f.fotoID, f.descripcion, f.fecha, f.me_gusta
```

```
from usuarios u -- Se mantienen todos los registros de esta tabla
```

```
-- Mantiene todos los usuarios aunque no tengan fotos
```

```
left join fotos f -- Esta tabla solo contiene coincidencias
```

```
on u.usuarioID = f.usuarioID;
```

```
-- 2. Mostrar todas las fotos, incluyendo las que no tienen comentarios.
```

```
select f.fotoID, f.descripcion, f.fecha, f.me_gusta,
```

```
c.comentarioID, c.comentario, c.fecha as fecha_comentario
```

```
from fotos f -- Se mantienen todos los registros de esta tabla
```

```
-- Mantiene todas las fotos aunque no tengan comentarios
```

```
left join comentarios c -- Solo contiene coincidencias
```

```
on f.fotoID = c.fotoID;
```

```
-- 3. Mostrar todos los usuarios, incluyendo aquellos que no han realizado ningún comentario.
```

```
select u.usuarioID, u.nombre, u.apellido,
```

```
c.comentarioID, c.comentario, c.fecha as fecha_comentario
```

```
from usuarios u -- Se mantienen todos los registros de esta tabla
```

```
-- Mantiene todos los usuarios aunque no tengan comentarios
```

```
left join comentarios c -- Solo contiene coincidencias
```

```
on u.usuarioID = c.usuarioID;
```

```
-- RIGHT JOIN
```

```
-- 1) Mostrar todas las fotos, incluso las que no están asociadas a ningún usuario
```

```
select
```

```
u.usuarioID, u.nombre, u.apellido, -- Datos del usuario pueden salir nulos si no hay relación
```

```
f.fotoID, f.descripcion, f.fecha, f.me_gusta -- Datos de las fotos siempre se muestran
from usuarios u
```

```
-- Se usa right join para mantener todas las fotos aunque no tengan usuario
right join fotos f on u.usuarioID = f.usuarioID;
```

```
-- 2) Mostrar todos los comentarios, incluso los que no están asociados a ninguna foto
```

```
select
```

```
    f.fotoID, f.descripcion, f.fecha, -- Datos de la foto, pueden salir nulos si no hay relación
```

```
    c.comentarioID, c.comentario, c.fecha as fecha_comentario -- Datos del comentario, siempre se muestran
```

```
from fotos f
```

```
-- right join mantiene todos los comentarios aunque no tengan foto válida
```

```
right join comentarios c on f.fotoID = c.fotoID;
```

```
-- 3) Mostrar todos los comentarios, incluso los que no están asociados a ningún usuario
```

```
select
```

```
    u.usuarioID, u.nombre, u.apellido, -- Datos del usuario, pueden ser nulos
```

```
    c.comentarioID, c.comentario, c.fecha as fecha_comentario -- Datos del comentario, siempre se muestran
```

```
from usuarios u
```

```
-- Se usa RIGHT JOIN para asegurar que todos los comentarios aparezcan aunque no tengan autor
```

```
right join comentarios c on u.usuarioID = c.usuarioID;
```

```
-- FULL OUTER JOIN
```

```
-- 1) Mostrar todos los usuarios y fotos, incluidas las que no tienen relación entre sí.
```

```
-- Usuarios que tienen o no tienen fotos
```

```
select
```

```
u.usuarioID, u.nombre, u.apellido,
```

```
f.fotoID, f.descripcion, f.fecha, f.me_gusta
```

```
from usuarios u
```

```
left join fotos f on u.usuarioID = f.usuarioID;
```

```
-- Fotos que tienen o no tienen usuario, por si hay fotos sin autor
```

```
select
```

```
u.usuarioID, u.nombre, u.apellido,
```

```
f.fotoID, f.descripcion, f.fecha, f.me_gusta
```



```
from usuarios u
right join fotos f on u.usuarioID = f.usuarioID;
```

-- 2) Mostrar todas las fotos y comentarios, incluidos los no relacionados entre sí

-- Fotos que pueden tener comentarios o no

```
select
f.fotoID, f.descripcion, f.fecha, f.me_gusta,
c.comentarioID, c.comentario, c.fecha as fecha_comentario
from fotos f
left join comentarios c on f.fotoID = c.fotoID;
```

-- Comentarios que pueden no estar relacionados con ninguna foto

```
select
f.fotoID, f.descripcion, f.fecha, f.me_gusta,
c.comentarioID, c.comentario, c.fecha as fecha_comentario
from fotos f
right join comentarios c on f.fotoID = c.fotoID;
```

-- 3) Mostrar todos los usuarios y comentarios, incluidos los no relacionados entre sí

-- Usuarios que pueden tener o no tener comentarios

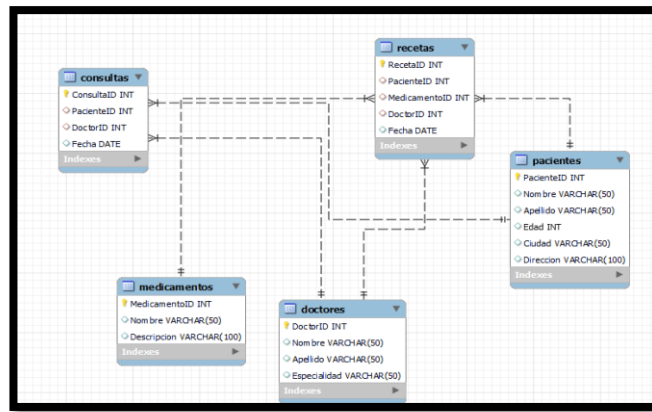
```
select
u.usuarioID, u.nombre, u.apellido,
c.comentarioID, c.comentario, c.fecha as fecha_comentario
from usuarios u
left join comentarios c on u.usuarioID = c.usuarioID;
```

-- Comentarios que pueden no tener usuario

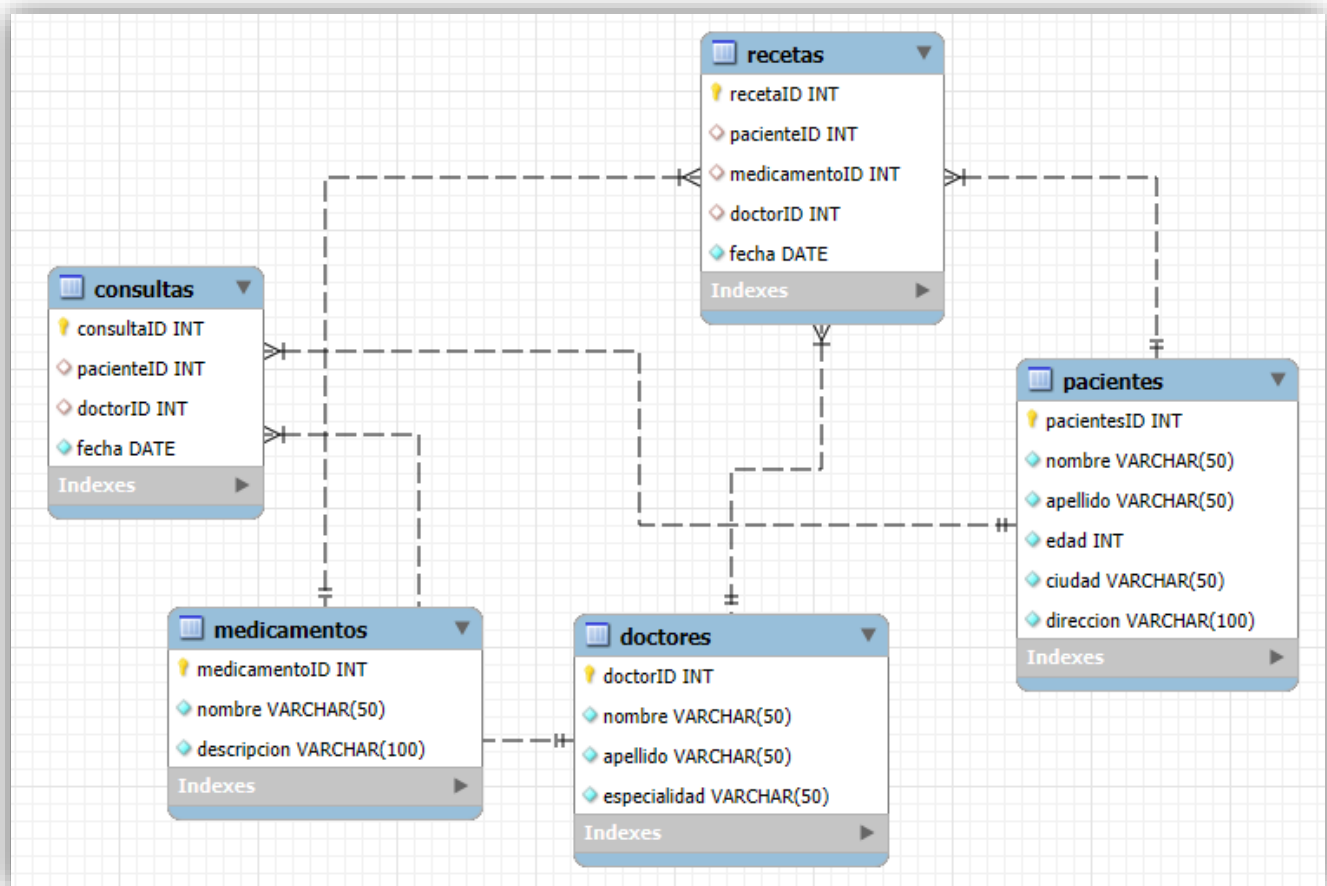
```
select
u.usuarioID, u.nombre, u.apellido,
c.comentarioID, c.comentario, c.fecha as fecha_comentario
from usuarios u
right join comentarios c on u.usuarioID = c.usuarioID;
```

Temática	INNER JOIN	LEFT JOIN	RIGHT JOIN	FULL OUTER JOIN
Clínica	Relacionar <b>pacientes</b> con <b>consultas</b> cuando ambos están registrados.	Mostrar todos los <b>pacientes</b> , incluyendo aquellos que aún no tienen <b>consultas</b> agendadas.	Mostrar todas las <b>consultas</b> , incluso aquellas que no están asignadas a un <b>paciente</b> .	Mostrar todos los <b>pacientes</b> y <b>consultas</b> , incluidas las que no están relacionadas entre sí.
	Relacionar <b>consultas</b> con <b>doctores</b> asignados.	Mostrar todas las <b>consultas</b> , incluyendo las que no tienen <b>doctor</b> asignado.	Mostrar todos los <b>doctores</b> , incluso aquellos que no tienen <b>consultas</b> asignadas.	Mostrar todas las <b>consultas</b> y <b>doctores</b> , incluso las no relacionadas.
	Relacionar <b>medicamentos</b> recetados con los <b>pacientes</b> que los reciben.	Mostrar todos los <b>pacientes</b> , incluyendo aquellos que no reciben <b>medicamentos</b> .	Mostrar todos los <b>medicamentos</b> , incluso los que no han sido recetados.	Mostrar todos los <b>pacientes</b> y <b>medicamentos</b> , incluso los no relacionados entre sí.
Curso	Relacionar <b>estudiantes</b> con <b>cursos</b> en los que están inscritos.	Mostrar todos los <b>estudiantes</b> , incluso aquellos que no están inscritos en ningún <b>curso</b> .	Mostrar todos los <b>cursos</b> , incluso aquellos que no tienen <b>estudiantes</b> inscritos.	Mostrar todos los <b>estudiantes</b> y <b>cursos</b> , incluso los no relacionados entre sí.
	Relacionar <b>cursos</b> con sus <b>instructores</b> .	Mostrar todos los <b>cursos</b> , incluyendo aquellos sin <b>instructor</b> asignado.	Mostrar todos los <b>instructores</b> , incluso aquellos que no tienen <b>cursos</b> a su cargo.	Mostrar todos los <b>cursos</b> y <b>instructores</b> , incluidos los no relacionados entre sí.
	Relacionar <b>estudiantes</b> con <b>lecciones completadas</b> .	Mostrar todos los <b>estudiantes</b> , incluso aquellos que no han completado ninguna <b>lección</b> .	Mostrar todas las <b>lecciones</b> , incluyendo las que no han sido completadas por ningún <b>estudiante</b> .	Mostrar todos los <b>estudiantes</b> y <b>lecciones</b> , incluso las no relacionadas entre sí.
Redes Sociales	Relacionar <b>usuarios</b> con sus <b>fotos</b> publicadas.	Mostrar todos los <b>usuarios</b> , incluyendo aquellos que no han subido ninguna <b>foto</b> .	Mostrar todas las <b>fotos</b> , incluso las que no están asociadas a ningún <b>usuario</b> .	Mostrar todos los <b>usuarios</b> y <b>fotos</b> , incluidas las que no tienen relación entre sí.
	Relacionar <b>fotos</b> con los <b>comentarios</b> recibidos.	Mostrar todas las <b>fotos</b> , incluyendo las que no tienen <b>comentarios</b> .	Mostrar todos los <b>comentarios</b> , incluso los que no están asociados a ninguna <b>foto</b> .	Mostrar todas las <b>fotos</b> y <b>comentarios</b> , incluidos los no relacionados entre sí.
	Relacionar <b>usuarios</b> con sus <b>comentarios</b> realizados.	Mostrar todos los <b>usuarios</b> , incluyendo aquellos que no han realizado ningún <b>comentario</b> .	Mostrar todos los <b>comentarios</b> , incluso los que no están asociados a ningún <b>usuario</b> .	Mostrar todos los <b>usuarios</b> y <b>comentarios</b> , incluidos los no relacionados entre sí.

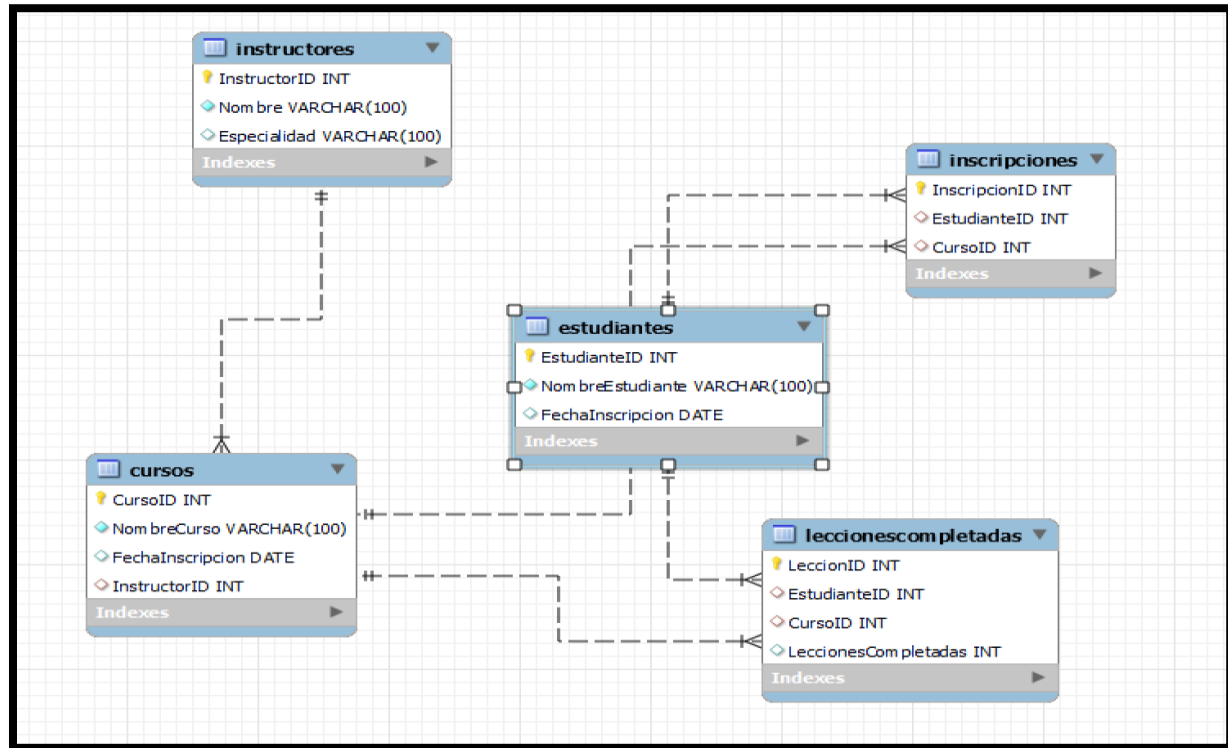
# CLÍNICA



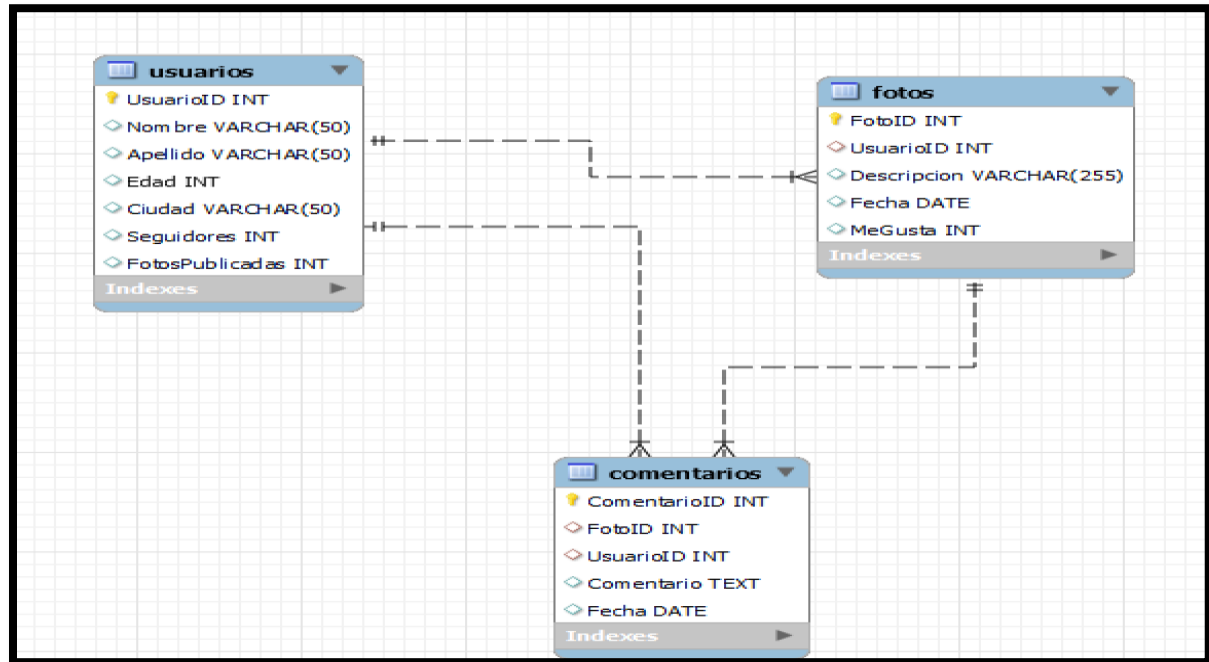
## DIAGRAMA



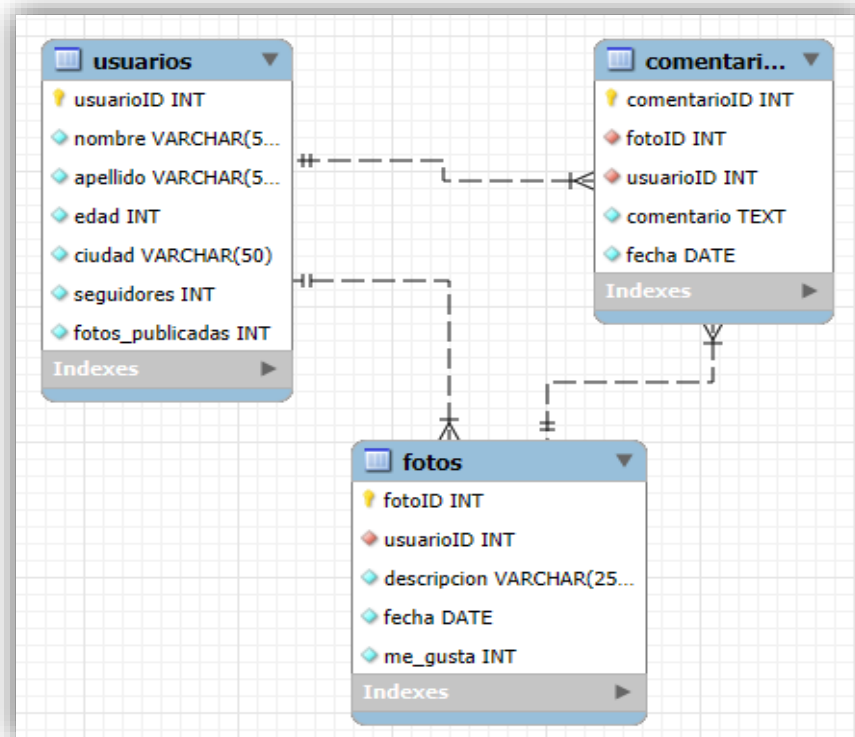
## Cursos ONLINE



## REDES SOCIALES



## DIAGRAMA



Subir información del informe y Script al Git hub

[https://github.com/NayelyAyol/TAREA\\_JOIN](https://github.com/NayelyAyol/TAREA_JOIN)