

Create database Colegio;

use Colegio;

Create table grupo

id_grupo int primary key,

nombre varchar (50),

);

Create table estudiante

id_estudiante int primary key,

nombre varchar (50),

edad int,

id_grupo int foreign key grupo (id_grupo)

;

Create table profesor

id_profesor int primary key,

Nombre varchar (50),

especialidad varchar (50)

;

1,400,877 vistas hace 11 horas #23 en Tendencias de música
BAD BUNNY - ALAMBRE PUA

CREATE TABLE cursos
ID-CURSO INT PRIMARY KEY,
NOMBRE VARCHAR(50),
ID-GRUPO INT FOREIGN KEY REFERENCES GRUPO(COD-GRUPO);

```
    );
CREATE TABLE material(
    id_material INT PRIMARY KEY,
    nombre VARCHAR(50)
);

```

```
    );  
Create Table estudiante_cursos  
id_estudiante int Foreign key references estudiante(id_estu),  
id_curso int Foreign key references curso (id_curso)  
);
```

```
    Create Table professor-material  
    id_professor int foreign key references professor(id_professor),  
    id_materia int foreign key references materia(id_materia)  
);
```

```
insert into grpo values (1,'10A');
```

Insert into estudiante values ('1', 'Juan', 15, 1);

insert into professor values (100, 'algebra', 1);

```
insert into Professor values(1,'carlos perez','matematicas');
```

insert into materia values(201,'mathematicus');

insert into estudiante_curso values (1, 100);

insert into Professor_materiel values (1, 201);

Select * from gruk

Select nombre from

select
from
left

left	Nombre:	estu
grow	Description:	
sele	clave	can
selk	PK	18.
sel		no
		ed
	fk	18.

sel	nombre
sel	descripc
sel	clave
sel	PK

50

5

2

Select * from grupos;

select nombre from grupos where id_grupo = 1;

select g.nombre, count(c.id_estudiante) As total_estudianck

from grupo g

left join estudiante e on g.id_grupo = e.id_grupo

group By g.nombre;

Select * from estudiante;

select nombre from estudiante where id_grupo = 1;

(Lección),

select nombre, edad from estudiante where edad > 15;

Select * from profesores;

select nombre from profesores where especialidad = 'matemáticas';

Select * from materias;

select * from materias where nombre like '%matem%';

Select * from profesor-materia;

select id_materia from profesor-materia where id_profesor = 1;

select nombre from curso where id_grupo = 1;

Select nombre from estudiante

where id_estudianck =

Select id_grupo from estudiante where id_estudiante = 1;

);

Select nombre from estudiante

Select e.nombre As estudiante, c.nombre As curso
from estudiante e
Join estudiante curso ec on e.id-estudiante = ec.id-estudiante
Join curso c on ec.id-curso = c.id-curso;

Select p.nombre As Profesor, m.nombre As materia
from profesor p
Join profesor-materia pm on P.id-profesor = pm.id-profesor
Join materia m on pm.id-materia = m.id-materia;

Select g.nombre As grupo, e.nombre As estudiante
from grupos g
Join estudiante e on g.id-grupo = e.id-grupo;

update estudiante set nombre = 'Juan Perez' where id-estudiante=1;

update grupos set nombre = 'IOB' where id-grupo=1;

delete from estudiante-curso where id-estudiante = 1

delete from estudiante where id-estudiante = 1

drop table estudiante-curso;

drop table estudiante;

Select e.nombre As estudiante, c.nombre As curso

nombre: curso

descripción: almacena información del curso

clave	campo	tipo de dato	tamaño	descripción
PK	id-curso	int		identificador del curso
	nombre	varchar	100	nombre del curso
FK	id-grupo	int		identificador del curso

nombre: estudiante_curso

descripción: tabla pivote entre estudiante y curso

clave	campo	tipo de dato	tamaño	descripción
FK	id-estudiante	int		identificador de estudiante
FK	id-curso	int		identificador de curso

nombre: profesor_materia

descripción: tabla pivote entre profesor y materia

clave	campo	tipo de dato	tamaño	descripción
FK	id-profesor	int		identificador de profesor
FK	id-materia	int		identificador de materia

nombre: materia

descripción: almacena información de materia

clave	campo	tipo de dato	tamaño	descripción
PK	id-materia	int		identificador de materia

Select e, nombre As nombre FROM e;

1

nombre: estudiante

descripción: Almacena la información de los estudiantes

clave	campo	tipo de dato	tamaño	descripción
PK	id-estudiante	int		identificador de estudiante
	nombre	varchar	100	nombre del estudiante
	edad	varchar	100	edad del estudiante
FK	id-grupo	int		identificador del grupo

nombre: grupo

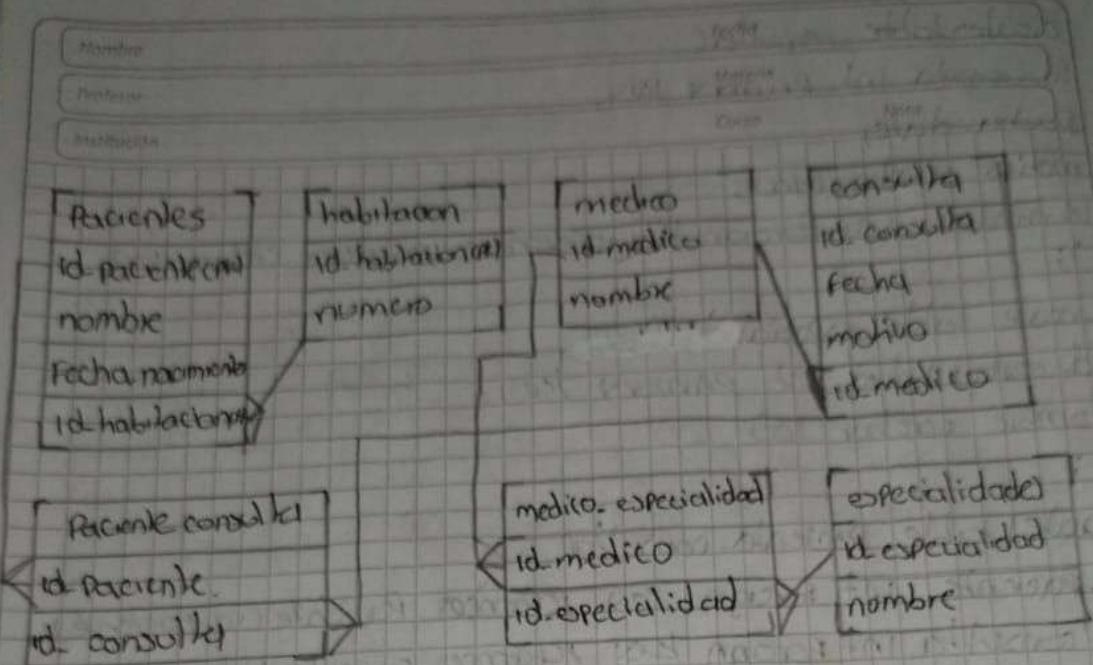
descripción: almacena información del grupo

clave	campo	tipo de dato	tamaño	descripción
PK	id-grupo	int		identificador de grupo
	nombre	varchar	100	nombre del grupo

nombre: profesor

descripción: almacena información del profesor

clave	campo	tipo de dato	tamaño	descripción
PK	id-profesor	int		identificador de profesor
	nombre	varchar	100	nombre del profesor
	especialidad	varchar	100	especialidad del profesor



Create database hospital;

use hospital;

Create table habitacion

`id_habitacion int Primary Key,`

`numero varchar (10);`

`;`

Create table Paciente(

`id_Paciente int Primary Key,`

`nombre varchar (30),`

`Fecha_nacimiento date,`

`id_habitacion int Foreign Key habitacion(id_habitacion)`

`;`

Create table medico(

`id_medicos int Primary Key,`

`nombre varchar (50),`

`especialidad varchar (30)`

`;`

```
create table consultaC  
id_consulta int primary key,  
fecha date,  
medico varchar(100),  
id_medico int foreign key references medicos (id-medico);  
);
```

```
Create Table especialidadC  
id_especialidad int primary key,  
nombre varchar(100)  
);
```

```
Create Table paciente_consultaC  
id_paciente int foreign key references paciente (id-paciente),  
id_consulta int foreign key references consulta (id-consulta)  
);
```

```
Create Table medico-especialidadC  
id_medico int foreign key references medico (id-medico),  
id_especialidad int foreign key references especialidad (id-especialidad)  
);
```

```
Insert into habitacion values (1, 'A101');
```

```
Insert into paciente values(1, 'Laura Gomez', '1990-04-23', 1);
```

```
Insert into medico values(1, 'Pedro Martinez', 'cardiologia');
```

```
Insert into consulta values(1001, '2025-07-01', 'dolor de pecho', 1);
```

```
Insert into especialidad values(201, 'cardiologia');
```

```
Insert into paciente_consulta values(1, 1001);
```

```
Insert into medico-especialidad values(1, 201);
```

select * from habilitacion;

select numero from habilitaciones where id-habilitacion = 1;

select h.numero , count(c.p.id-paciente) AS pacientes
from habilitacion h

left join paciente p on h.id-habilitacion = p.id-habilitacion
group by h.numero;

select * from paciente;

select nombre from paciente where id-habilitacion = 1;

select * from medicos;

select nombre from medicos where especialidad = 'cardiologo';

select * from consultas;

select fecha from consultas where fecha >= '2023-07-01';

select * from especialidad;

select e.nombre , count(m.id-medico) AS total-medicos

from especialidad e

left join medico-especialidad me on e.id-especialidad = me.id-especialidad
group by e.nombre;

select nombre from especialidad where nombre like '%cardio%'

select * from paciente-consulta;

select id-consulta from paciente-consulta where id-paciente = 1;

select * from medico-especialidad;

select id-especialidad from medico-especialidad where id-medico = 1;

Select nombre from Paciente
where id_paciente in (

select id_paciente from paciente_consulta

);

Select * from consulta where id_medico in (

Select id_medico from medico where especialidad='cardiologia'
);

Select * from habilidades

where id_habilidad in (

Select id_habilidad from paciente

);

Select p.nombre As Paciente, h.numero As Habilidad

from paciente p

Join habilidades h on p.id_habilidad = h.id_habilidad;

Select m.nombre As medico, e.nombre As especialidad

from medico m

Join medico_especialidad me on m.id_especialidad = e.id_especialidad;

Update habilitacion set numero = 'A102' where id_habilidad = 1;

Update paciente set nombre = 'laura G.' where id_paciente = 1;

Delete from especialidad where id_especialidad = 201;

Drop table consulta;

selecionar nombre from Paciente
where id-paciente = 6

nombre: medico				
Descripción: almacena información del medico				
Clave	Campo	Tipo de dato	Largo	Descripción
PK	id-medico	int		Identificador del medico
	nombre	varchar	100	Nombre del medico

nombre: especialidad				
Descripción: almacena información de especialidades				
Clave	Campo	Tipo de dato	Largo	Descripción
PK	id-especialidad	int		Identificador de especialidad
	nombre	varchar	100	Nombre de la especialidad

nombre: paciente_consulta				
Descripción: tabla pivote entre Paciente y consulta				
Clave	Campo	Tipo de dato	Largo	Descripción
FK	id-paciente	int		Identificador del paciente
FK	id-consulta	int		Identificador de consulta

nombre: medico_especialidad				
Descripción: tabla pivote entre medico y especialidad				
Clave	Campo	Tipo de dato	Largo	Descripción
FK	id-medico	int		Identificador de medico
FK	id-especialidad	int		Identificador de especialidad

se
wr

v

nombre: paciente

descripcion: almacena informacion de paciente

clave	campo	tipo de dato	tamaño	descripcion
PK	id-paciente	int		identificador de paciente
	nombre	varchar	100	nombre del paciente
	fecha	date		fecha de nacimiento
FK	id-habitacion	int		identificador de habitacion

nombre: habitacion

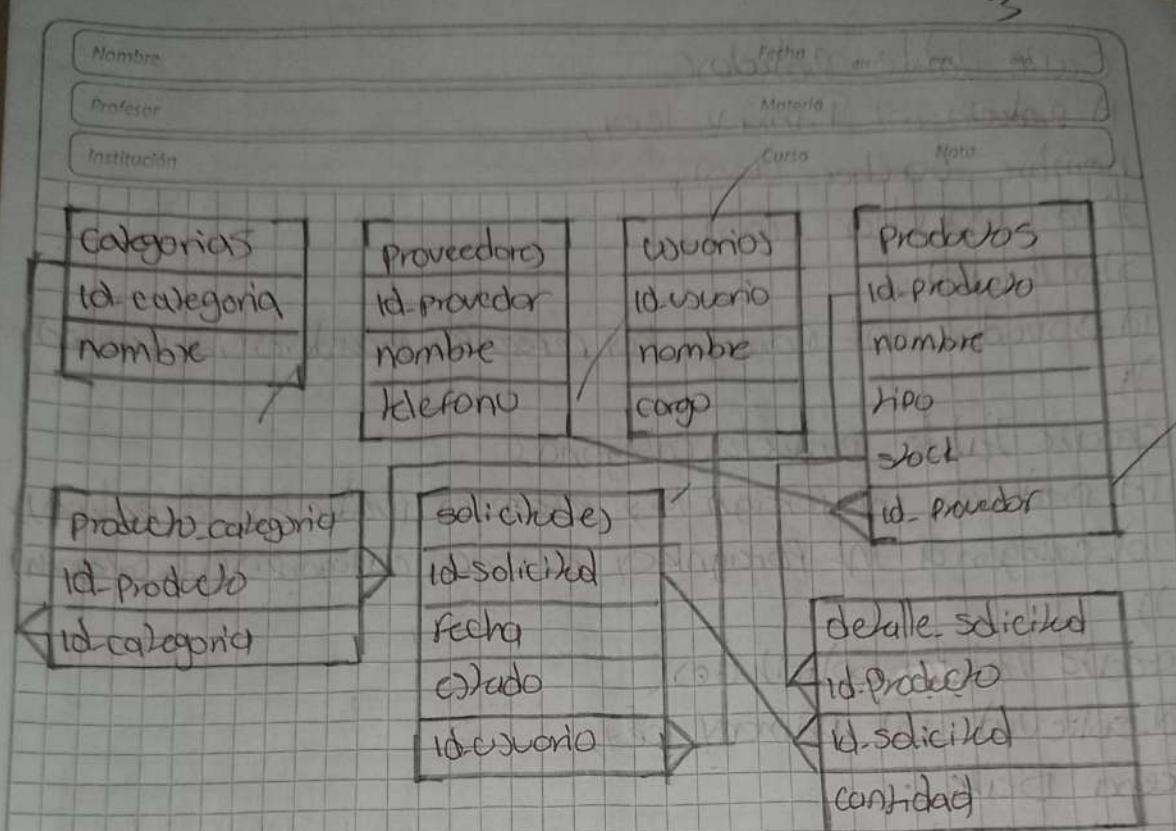
descripcion: almacena informacion de la habitacion

clave	campo	tipo de dato	tamaño	descripcion
PK	id-habitacion	int		identificador de la habitacion
	numero	varchar	100	numero de la habitacion

nombre: consulta

descripcion: almacena informacion de los consultas

clave	campo	tipo de dato	tamaño	descripcion
PK	id-consulta	int		identificador de consulta
	fecha	date		fecha de la consulta
	motivo	varchar	100	motivo de la consulta
FK	id-medico	int		identificador de medico



Create database inventario;

Use inventario;

Create table categorias(

id_categoria int Primary key,

nombre varchar (100)

);

Create table proveedor(

id_proveedor int Primary key,

nombre varchar (100),

telefono varchar (100)

);

Create table usuarios(

id_usuario int Primary key,

nombre varchar (100),

correo varchar (100)

);

Create table Productos
Id_producto int Primary key,
Nombre varchar (100),
Tipo varchar (100),
Stock int,
Id_Proveedor int foreign key references Proveedores (Id_Proveedor)

i) Create table Producto_Categoría

Id_Producto int foreign key references Productos (Id_Producto),
Id_Categoría int foreign key references Categorías (Id_Categoría)

);

Create table Solicitudes

Id_Solicitud int Primary key,
Fecha Date,
Estado Varchar (100),
Id_Usuario int foreign key references Usuarios (Id_Usuario)

);

Create table Detalle_Solicitud

(Id_Solicitud int foreign key references Solicitudes (Id_Solicitud),
Id_Producto int foreign key references Productos (Id_Producto),
Cantidad int
)

Insert into Categorías values (1, 'Papelera');

Insert into Proveedores values (1, 'Distribuidora S.A.', '3000');

Insert into Usuarios values (1, 'Loca', 'Coordinadora');

Insert into Productos values (1, 'Rellena Papel', 'Oficina'), 100, 1);

Insert into Producto_Categoría values (1, 1);

Insert into Solicitudes values (1, '2025-02-05', 'pendiente', 1);

Insert into Detalle_Solicitud values (1, 1, 20);

select * from Producto;

Select nombre from Producto where stock < 50;

select * from Categorias;

select nombre from Categorias where idCategoria in (

select idCategoria from Producto_Categoria

);

select * from Proveedores;

select nombre from Proveedores where telefono like '%300%';

select nombre from Proveedores where id_Proveedor in (

select id_Proveedor from Productos

);

select * from Usuarios;

select nombre from usuario where cargo = 'Coordinador';

select nombre from usuarios where id_usuario in (

select id_usuario from solicitudes

);

select * from solicitudes

select fecha from solicitudes where estado = 'Pendiente';

select * from Producto_Categorias;

select id_producto from Producto_Categorias where id_categoria = 1;

select * from detalle_solicitud

select id_producto from detalle_solicitud where cantidad > 10;

select nombre from Productos

where id_producto in c

Select id_producto from detalle_solicitud

where id_solicitud in c

select id_solicitud from solicituds where estado = 'pendiente'

1)

1)

select nombre from usuarios

where id_usuario in c

select id_usuario from solicitudes

1)

Select p.nombre, pr.nombre As proveedor

From Productos P

Join Proveedores pr on P.id_Proveedor = pr.id_Proveedor;

Select s.id_solicitud, u.nombre As solicitante

From solicitudes s

Join usuarios u on s.id_usuario = u.id_usuario;

Update Productos set stock = stock - 20 where id_producto = 1;

Update usuarios set cargo = 'Jefe de compras' where id_usuario = 1;

delete from solicitudes where id_solicitud = 1;

delete from Proveedores where id_Proveedor = 1;

drop table solicitudes;

drop table Proveedores;

3

nombre: solicitudes

Descripción: almacena la información de las solicitudes

Clave	campo	tipo de dato	tamaño	descripción
PK	id_solicitud	int		identificador de solicitud
	fecha	date		fecha de solicitud
	estado	varchar	100	estado de solicitud
FK	id_usuario	int		identificador de usuario

nombre: productos

Descripción: almacena la información de producto

clave	campo	tipo de dato	tamaño	descripción
PK	id_producto	int		identificador del producto
	nombre	varchar	100	nombre del producto
	tipo	varchar	100	tipo del producto
	stock	int		stock del producto
FK	id_proveedor	int		identificador del proveedor

nombre: proveedores

Descripción: almacena información de los proveedores

clave	campo	tipo de dato	tamaño	descripción
PK	id_proveedor	int		identificador de proveedor
	nombre	varchar	100	nombre del proveedor
	telefono	varchar	100	telefono del proveedor

Select nombre from Productos
 where id_producto in c
 Select id_producto from detalle_solicitud

nombre: categoria			
Descripción: almacena la información de categoría			
clave	campo	tipo de dato	tamaño
PK	id_categoria	int	descripción

nombre: categoria
identificador de categoría

nombre: nombre
varchar 100 nombre de la categoría

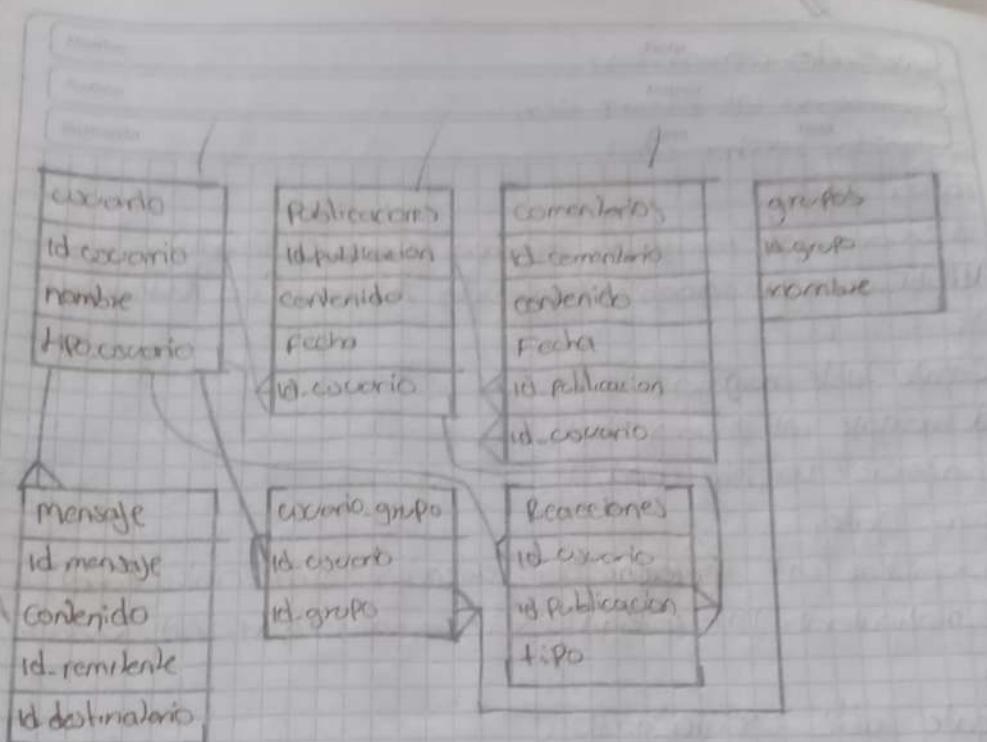
nombre: usuario			
Descripción: almacena información de los usuarios			
clave	campo	tipo de dato	tamaño
PK	id_usuario	int	descripción

clave: id_usuario
identificador de usuario

nombre: nombre
varchar 100 nombre del usuario

nombre: Producto_Categoría			
Descripción: tabla pivote entre Producto_Categoría			
clave	campo	tipo de dato	tamaño
FK	id_Producto	int	descripción
FK	id_categoria	int	identificador de producto
			identificador de categoría

nombre: detalle_solicitud			
Descripción: tabla pivote entre Productos y solicitudes			
clave	campo	tamaño	tipo de dato
FK	id_Producto	int	descripción
FK	id_solicitud	int	identificador de producto
			identificador de solicitud



Create Table Usuario(

`id_usuario int Primary Key,`

`nombre Varchar (100),`

`tipo_usuario Varchar (200)`

`);`

Create Table Publicacion(

`id_Publicacion int Primary Key,`

`contenido Varchar (100),`

`fecha Date,`

`id_usuario int Foreign Key reference) Usuario (id_usuario)`

`);`

Create Table Grupo(

`id_grupo int Primary Key,`

`nombre Varchar (100).`

`);`

Create table comentarios (

id_comentario int primary key,

contenido varchar(100),

fecha date,

id_usuario int foreign key references usuario(id_usuario),

id_publicacion int foreign key references publicacion(id_publicacion);

);

Create table mensajes (

id_mensaje int primary key,

contenido varchar(100),

fecha Date,

id_remitente int foreign key references usuario(id_usuario),

id_destinatario int foreign key references usuario(id_usuario);

);

Create table usuario_grupo (

id_usuario int foreign key references usuario(id_usuario),

id_grupo int foreign key references grupo(id_grupo);

);

Create table reacciones (

id_usuario int foreign key references usuario(id_usuario),

id_publicacion foreign key references publicaciones(id_publicacion),

tipo varchar(100)

);

Insert into usuarios values(1,'andrea','estudiante');

Insert into grupo values(1,'matematicas');

Insert into publicaciones values(1,'Hda grupo','2025-07-01',1);

Insert into comentarios values(1,'Hda andrea','2025-07-01',1,1);

Insert into usuario_grupo values(1,1);

insert into reacciones values (1, 1, 'Like');

insert into mensajes values (1, 'ENOT1', '2013-07-01', 1, 1);

select * from usuarios;

select nombre from usuarios where tipo_usuario = 'Joven';

select * from publicaciones;

select contenido from publicaciones where id_publicacion in

select id_publicacion from reacciones where tipo = 'Like';

);

select * from comentarios;

select contenido from comentarios where id_usuario = 1;

select * from grupos;

select nombre from grupos where id_grupo in

select id_grupo from usuario_grupo where id_usuario = 1

);

select * from mensajes;

select contenido from mensajes where id_destinatario = 1;

select * from usuario_grupo;

select id_usuario from usuario_grupo where id_grupo = 1;

select * from reacciones;

select tipo from reacciones where id_publicacion = 1;

select id_publicacion from reacciones where id_usuario = 1;

Select nombre from usuarios

where id_usuario in

Select id_usuario from reacciones where id_publicacion = 1

i)

Select id_publicacion from Publicacion

where id_publicacion in

Select distinct id_publicacion from comentarios

ii)

Select nombre from grupo

where id_grupo in

Select id_grupo from usuario_grupo

group by id_grupo having count(*) > 1

iii)

Select p.contenido, u.nombre As autor

From publicaciones p

join usuarios u on p.id_usuario = u.id_usuario;

Select u.nombre, g.nombre As grupo

From usuario u

Join usuario_grupo ug on u.id_usuario = ug.id_usuario

Join grupos g on ug.id_grupo = g.id_grupo;

Delete from publicaciones where id_publicacion = 1;

Drop table reacciones;

Drop table Publicaciones;

nombre	Comentarios
descripcion	alma
clave	Campo
PK	id_comentario
cont	
ta	
FK	
FL	
FK	
nom	
co	

6
Select nombre from usuarios
where id_usuario = 1

4

Nombre: Comentarios					
Descripción: Almacena información de los comentarios					
Clave	Campo	Tipo de dato	Tamaño	Descripción	
PK	id_comentario	int		Identificador de comentario	
	contenido	varchar	100	Contenido del comentario	
	fecha	Date		Fecha del comentario	
FK	id_publicacion	int		Identificador de publicación	
	id_usuario	int		Identificador del usuario	

Nombre: Publicaciones					
Descripción: Almacena información de publicaciones					
Clave	Campo	Tipo de dato	Tamaño	Descripción	
PK	id_publicacion	int		Identificador de publicación	
	contenido	varchar	100	Contenido de la publicación	
	fecha	Date		Fecha de publicación	
FK	id_usuario	int		Identificador de usuario	

Nombre: Mensaje					
Descripción: Almacena información de mensaje					
Clave	Campo	Tipo de dato	Tamaño	Descripción	
PK	id_mensaje	int		Identificador de mensaje	
	contenido	varchar	100	Contenido de mensaje	
FK	id_destinatario	int		Identificador de destinatario	

Select nombre from usuario
 where id_usuario = 1
 inner join reaccion on id_publicacion = 1

nombre: usuario

descripción: almacena información del usuario

clave	campo	tipo de dato	tamaño	descripción
PK	id_usuario	int		identificador del usuario
	nombre	varchar	100	nombre del usuario
	id_publicacion	varchar	100	Id de usuario

nombre: grupo

descripción: almacena la información del grupo

clave	campo	tipo de dato	tamaño	descripción
PK	id_grupo	int		identificador del grupo
	nombre	varchar	100	nombre del grupo

nombre: usuario_grupo

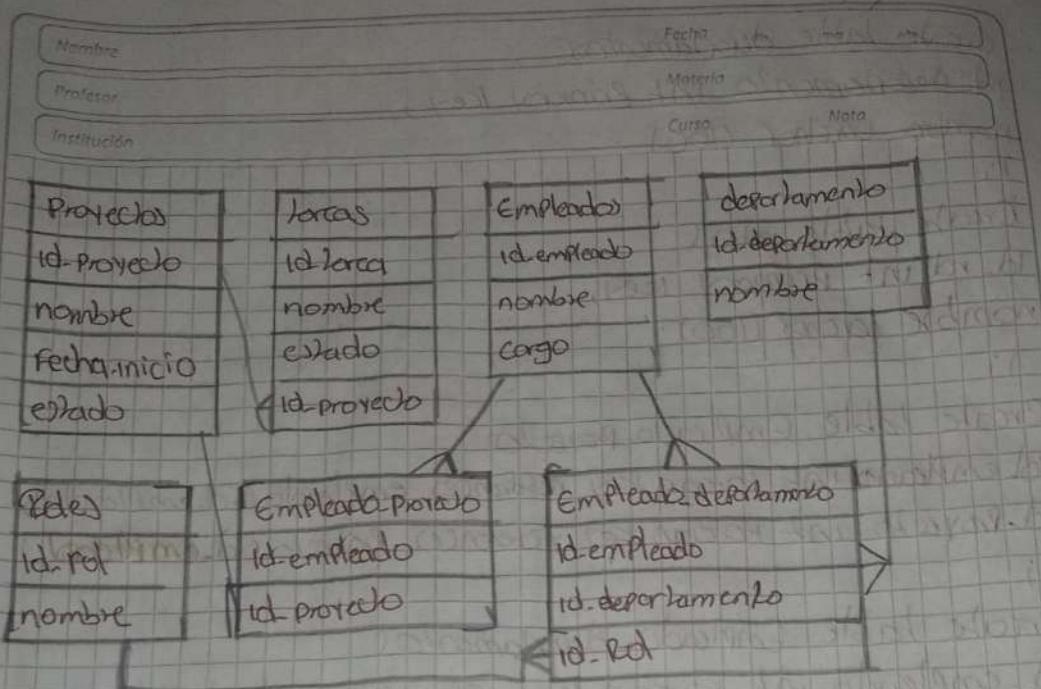
descripción: tabla pivote

clave	campo	tipo de dato	tamaño	descripción
PK	id_grupo	int		identificador de grupo
PK	id_usuario	int		identificador de usuario

nombre: reaccion

descripción: tabla pivote

clave	campo	tipo de dato	tamaño	descripción
FK	id_usuario	int		identificador de usuario
FK	id_publicacion	int		identificador de publicación



Create table Proyectos

```
id_proyecto int primary key,
nombre varchar(100),
fecha_inicio date,
estado varchar(100)
);
```

Create table lareas

```
id_larea int primary key,
nombre varchar(100),
estado varchar(100),
id_proyecto int foreign key references Proyecto(id_proyecto)
```

);

Create table Empleados

```
id_empleado int primary key,
nombre varchar(100),
cargo varchar(100)
```

);

Create table Departamento
id_departamento int primary key,
nombre varchar(100)
;
Create table roles
id_rd int primary key,
nombre varchar(100).
;
Create Table Empleado-Proyecto
id_empleado int foreign key references empleado (id_empleado),
id_proyecto int foreign key references proyecto (id_proyecto);
;
Create Table Empleado-departamento
id_empleado int foreign key references empleado (id_empleado),
id_departamento int foreign key references departamento (id_departamento),
id_rd int foreign key references roles (id_rd);
;
Insert into proyectos values (1,'sistema contable','2025-06-01','Achis');
Insert into empleado values (1,'Santiago Gordo','Desarrollador');
Insert into departamento values (1,'TI');
Insert into roles values (1,'Lider tecnico');
Insert into roles values (1,'base de datos','pendiente', 1);
Insert into Empleado-Proyecto (1,1);
Insert into empleado-departamento (1, 1, 1);

select * from proyectos;

select nombre from proyectos where estado = 'Activo';

select nombre from proyectos where fecha_inicio < '2025-07-01';

select * from tareas;

select count(*) As tareas_total from tareas where id_proyecto = 1;

select * from empleados;

select nombre from empleados where cargo = 'Desarrollador';

select nombre from empleados where id_empleado IN (

select id_empleado from empleado_proyecto where id_proyecto = 1;

);

select * from departamento;

select nombre from departamento where id_departamento = 1;

select * from roles;

select nombre from roles where nombre like '%.Lider%');

select nombre from roles where id_rol IN (

select id_rol from empleado_departamento

);

select * from empleado_proyecto;

select id_proyecto from empleado_proyecto where id_proyecto > 1;

select * from empleado_departamento;

select id_empleado from empleado_departamento where id_departamento = 1;

select nombre from empleado

where id-empleado in (

select id-empleado from empleado_proyecto where id_proyecto = 1
);

select nombre from proyectos

where id_proyecto in (

select id_proyecto from tareas where estado = 'pendiente'

);

select nombre from departamento

where id-departamento in (

select id_departamento from empleado_departamento

);

select e.nombre As empleado, p.nombre As Proyecto
from empleado e

Join empleado_proyecto ep on e.idempleado = ep.id_empleado
Join proyectos P on ep.id_proyectos = p.id_proyectos;

select P.nombre As Proyecto, t.nombre As Tarea, t.estado
From proyectos P
Join tareas t on p.id_proyecto = t.id_proyecto;

Update tareas set estado = 'completada' where id_tarea = 1;

Update proyectos set estado = 'Analizado' where id_proyecto = 1;

Delete from tareas where id_tarea = 1;

Delete from proyectos where id_proyecto = 1;

Drop table tareas;

Drop proyectos;

selec nombre from empleados
where id_empleado = 10

Nombre: Proyectos				
Descripción: Almacena información de proyectos				
Clave	Campo	Tipo de dato	Largo	Descripción
PK	id_proyecto	int		Identificador de proyecto
	nombre	varchar	100	Nombre del proyecto
	fecha	date		Fecha del proyecto
	estado	varchar	100	Estado del proyecto

Nombre: Tareas				
Descripción: Almacena información de tareas				
Clave	Campo	Tipo de dato	Largo	Descripción
PK	id_tarea	int		Identificador de tarea
	nombre	varchar		Nombre de la tarea
	estado	varchar		Estado de la tarea
EK	id_proyecto	int		Identificador del proyecto

Nombre: Empleados				
Descripción: Almacena información de empleado				
Clave	Campo	Tipo de dato	Largo	Descripción
PK	id_empleado	int		Identificador de empleado
	nombre	varchar	100	Nombre del empleado
	cargo	varchar	100	Cargo del empleado

selecion nombre from empleado
where id_empleado in
select id_empleado from empleado_proyecto where id_proyecto = 1

nombre_departamento

descripcion: almacena informacion de departamento

Clave	Campo	Tipo de dato	Nombre	Descripcion
PK	id_departamento	varchar	Identificador del departamento	
	nombre			Nombre del departamento.

nombre - rol

descripcion: almacena informacion de roles

Clave	Campo	Tipo de dato	Nombre	Descripcion
PK	id_rol	varchar	Identificador del rol	
	nombre	int		Nombre del rol

nombre: empleado_proyecto

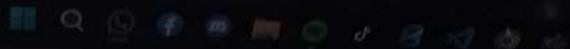
descripcion tabla pivote

Clave	Campo	Tipo de dato	Nombre	Descripcion
FK	id_empleado	int	Identificador de empleado	
FK	id_proyecto	int	Identificador de proyecto	

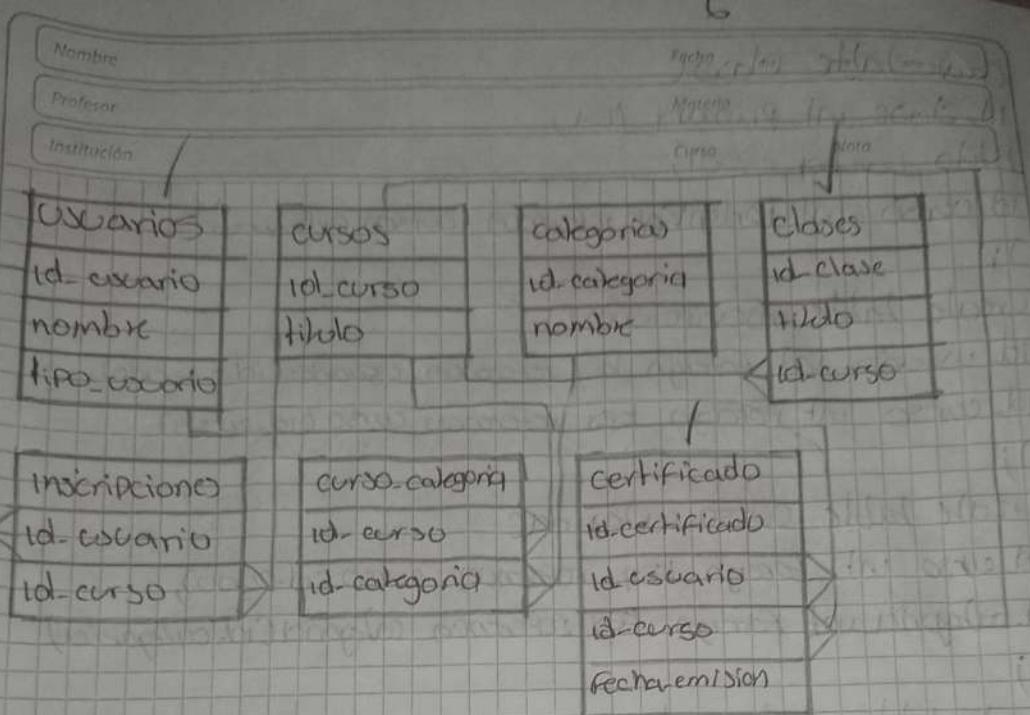
nombre: empleado ->

descripcion tabla pivote

Clave	Campo	Tipo de dato	Nombre	Descripcion
FK	id_empleado	int	Identificador de empleado	
FK	id_departamento	int	Identificador del departamento	



ASUS Vivobook



Create database CursosOnline;

Use cursosOnline;

Create Table usuarios

id_usuario int Primary Key,

nombre varchar (100),

tipo_usuario varchar (100)

);

Create Table cursos

id_curso int Primary Key

título varchar (100)

);

Create Table categorías

id_categoria int Primary Key

nombre varchar (400)

);

Create Table Clases(

id_clase int Primary key,

titulo varchar (400),

id_curso int Foreign Key references cursos (id_curso)

)

Create Table Inscripciones(

id_usuario int Foreign Key references usuarios (id_usuario),

id_curso int Foreign Key references curso (id_curso)

)

Create Table Curso-Categorias(

id_curso int Foreign Key references curso (id_curso),

id_categoria int Foreign Key references categorias (id_categoria)

)

Create Table Certificados(

id_certificado int Primary key,

id_usuario int Foreign Key references usuarios (id_usuario),

id_curso int Foreign Key references cursos (id_curso),

fecha_emision date,

)

Insert into usuarios values (1, 'Carlos Perez', 'Colombiano')

Insert into cursos values (1, 'Python desde cero')

Insert into categorias values (1, 'Programacion')

Insert into clases values (1, 'Introducción a Python', 1)

Insert into inscripciones values (1, 1)

Insert into curso-categorias values (1, 1)

Insert into certificados values (1, 1, 1, '2025-07-01')

Select * from usuarios

Select nombre from usuarios

Select * from cursos

Selec

Selec

s

)

Selec

sel

Selec

sc

)

Selec

s

)

Nombre:	Car
Descripción:	Al
Clave:	Com
PK:	id_c
+1:	

Nombre:	Co
Descripción:	Al
Clave:	(
PK:	id
+1:	

Nombre:	Ca
Descripción:	Al
Clave:	(
FK:	
FK:	

Nombre:	Oce
Descripción:	Al
Clave:	(
FK:	
FK:	

select * from usuarios;

select nombre from usuarios where tipo_usuario = 'estudiante';

select * from curso;

select titulo from curso where titulo like '%Python%';

select titulo from cursos where id_cursos in (

select id_curso from inscripciones where id_usuario = 1

)

select * from categoria;

select nombre from categoria where nombre like '%Programación%';

select * from clase;

select titulo from clases where id_clase > 1;

select count(*) As total_clases from clases where curso = 7;

select * from inscripciones;

select id_curso from inscripciones where id_usuario = 1;

select * from curso_categoria;

select id_curso from curso_categoria where id_categoria = 1;

select * from certificado;

select id_usuario from certificados where fecha_emision

Between '2025-07-01' And '2025-07-09';

```
SELECT u.nombre AS usuario, c.titulo AS curso  
FROM usuario u  
JOIN inscripciones i ON u.id_usuario = i.id_usuario  
JOIN cursos c ON i.id_curso = c.id_curso;  
  
SELECT u.nombre AS usuario, cu.titulo AS curso, ce.fecha_mision  
FROM certificados ce  
JOIN usuario u ON ce.id_usuario = u.id_usuario  
JOIN cursos cu ON ce.id_curso = cu.id_curso  
  
SELECT titulo FROM cursos  
WHERE id_curso IN  
    (SELECT id_curso FROM inscripciones WHERE id_usuario = 1)  
;  
SELECT nombre FROM usuarios  
WHERE id_usuario IN  
    (SELECT id_usuario FROM certificados)  
;  
SELECT DISTINCT cat.nombre  
FROM categorias cat  
JOIN curso_categorias cc ON cat.id_categoria = cc.id_categoria  
WHERE cc.id_curso IN  
    (SELECT id_curso FROM inscripciones WHERE id_usuario = 1)  
;  
UPDATE cursos SET titulo = 'Python Profesional' WHERE id_curso = 1;  
DELETE FROM cursos WHERE id_curso = 1;  
DROP TABLE cursos;
```

selección de nombre del curso, título del curso
número de identificación del curso

Nombre: curso				
Descripción: almacena información del curso				
Clave	campo	Tipo de dato	Largo	Descripción
PK	id-curso	int		Identificador del curso
	título	varchar	100	Nombre del curso

Nombre: categoría				
Descripción: almacena información de las categorías				
Clave	campo	Tipo de dato	Largo	Descripción
PK	id-categoría	int		Identificador de categoría
	nombre	varchar	100	Nombre de la categoría

Nombre: inscripciones				
Descripción: tabla pivote				
Clave	campo	Tipo de dato	Largo	Descripción
FK	id-curso	int		Identificador del curso
FK	id-curso	int		Identificador del curso

Nombre: curso-categoría				
Descripción: almacena información de curso x categoría				
Clave	campo	Tipo de dato	Largo	Descripción
FK	id-curso	int		Identificador del curso
FK	id-categoría	int		Identificador de categoría

Select
from
Join
Join
Select
from
Join
Join

se
w'

monitc. certificado

Nombre: Certificado
Descripción: almacena información de certificados

Descripción		tipo de dato	tamaño	descripción
clave	campo	char de	dato	identificador de certificado
PK	id-certificado	int		identificador de usuario
FK	id-curso	int		identificador de curso
FK	id-curso	int		fecha emisión del certificado
	fecha-emisión	date		

nombre: cuatro

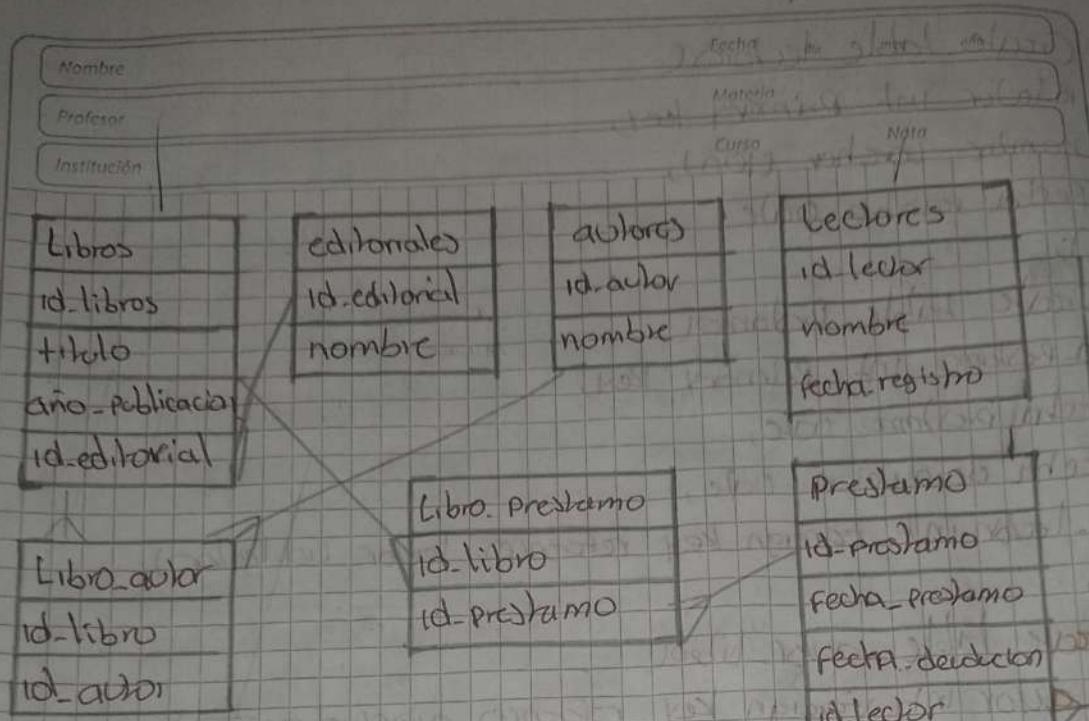
Descripción: almacena información de usuario

Clave	Campo	Tipo de dato	Leyenda	Descripción
PK	id_vacuno	int		Identificador del vacuno
	nombre	varchar	(100)	Nombre del vacuno
	tipo_vacuno	varchar	100	Tipo de vacuno

Nombre: Clase)

Descripción:almacen de información de clase

Clave	campo	tipo de dato	tamaño	descripción
PK	id-clase	int		Identificador de Clase
	nombre	varchar	100	nombre de la clase
FK	id-curso	int		Identificador de curso



Create database biblioteca;

use biblioteca

Create table editorial(

id_editorial int primary key,

nombre varchar (100)

)

Create table libros(

id-libro int primary key,

título varchar (100),

año_publicacion int,

id_editorial int foreign key references editorial (id_editorial)

)

Create table autores(

id-autor int primary key,

nombre varchar (100)

)

Create Table lectores

id-lector int primary key,
nombre varchar (100),
fecha-registro date

);

Create Table Prestamo

id-prestamo int primary key,
fecha-prestamo date,
fecha-devolucion date,
id-lector int foreign key references lector (id-lector)

);

Create Table autor-libro

id-autor int foreign key references autor (id-autor),
id-libro int foreign key references libro (id-libro)

);

Create Table libro-Prestamo

id-libro int foreign key references libro (id-libro),
id-Prestamo int foreign key references Prestamo (id-prestamo)

);

Insert into editorial values (1, 'Editorial Planeta');

Insert into libros values (1, 'Cien años de soledad', 1967, 1);

Insert into autores values (1, 'Gabriel García Márquez');

Insert into lectores values (1, 'María Rodríguez', '2024-01-15');

Insert into Prestamo values (1001, '2025-07-01', '2025-07-10', 1);

Insert into Autor-Libro values (1, 1);

selección de libros;

selección libro de libros where año-publicación < 2000;

selección libro de libros where id-editorial = 1;

selección de autores;

selección a.nombre, count(cel.id.libro) As libros

from autores a

join autor-libro al on a.id.autor = al.id.autor

group by a.nombre;

selección e from editoriales;

selección nombre from editoriales id.editoriales = 1;

selección * from lectores;

selección nombre from lectores where fecha-registro >='2024-01-01';

selección nombre from lectores where id-lector in (

selección id-lector from prelamo

);

selección * from autor-libro;

selección id-libro from autor-libro where id-autor = 1;

selección id-autor from autor-libro where id-libro = 1;

selección * from libro-prelamo;

selección id-Prelamo from libro-prelamo where id-libro = 1;

selección id-libro from libro-prelamo where id-Prelamo = 1001;

Select l.titulo, e.nombre AS editorial
From libros l
Join editorial e on l.id_editorial = e.id_editorial;
Select a.nombre AS autor, l.titulo AS libro
From autores a
Join Autor-libro al on a.id_autor = al.id_autor
Join libro l on al.id_libro = l.id_libro;
Select DISTINCT a.nombre
From autores a
Join autor-libro al on a.id_autor = al.id_autor
Where al.id_libro IN (

Select id-libro From libro_prestamo
);

Select nombre from lectores
Where id_lector in (

Select id_lector from prestamo
);

Update lectores set nombre = 'monica D' where id_lector = 1;

Update prestamo set fecha_devolucion = '2025-07-09' where id_prestamo > 100;

Delete from libro_prestamo where id_libro = 1;

Delete from Prestamo where id_prestamo = 1;

Drop table libro_prestamo;

Drop table Prestamo;

Select 1. titulo, e nombre As editorial

From

Join

Select

From

Join

Join

Select

From

Join

Where

Select

);

Select

Where

Select

Update

Update

Delete

Delete

Op

Op

nombre: editorial

descripcion: almacena informacion del editorial

clave	campo	tipo de dato	tamano	descripcion
PK	id-editorial	int		identificador del editorial
	nombre	varchar	100	nombre de la editorial

nombre: autor

descripcion: almacena informacion del autor

clave	campo	tipo de dato	tamano	descripcion
PK	id-autor	int		identificador del autor
	nombre	varchar	100	nombre del autor

nombre: libro-autor

descripcion: tabla pivote

clave	campo	tipo de dato	tamano	descripcion
FK	id-libro	int		identificador del libro
FK	id-autor	int		identificador del autor

nombre: libro_Prestamo

descripcion: tabla pivote

clave	campo	tipo de dato	tamano	descripcion
FK	id-libro	int		identificador del libro
FK	id-Prestamo	int		identificador del Prestamo

Select libro, c.nombre As editorial
From libros

7

6

nombre: Prestamo

descripcion: almacena informacion de prestamo

clave	campo	tipo de dato	temario	descripcion
PK	id_prestamo	int		identificador del prestamo
	fecha_prestamo	date		fecha prestamo
	fecha_devolucion	date		fecha devolucion
FK	id_lector	int		identificador del lector

nombre: Libro

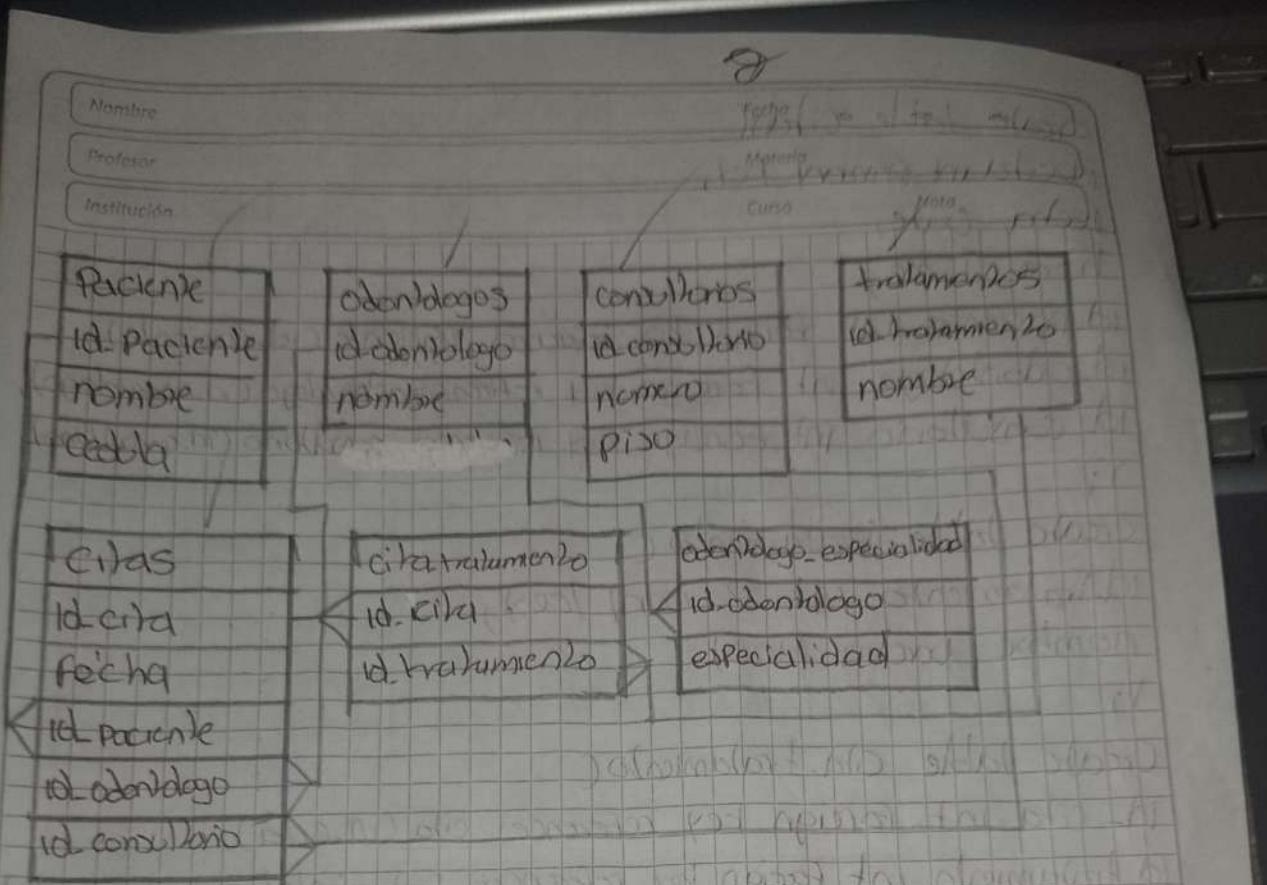
descripcion: almacena informacion del libro

clave	campo	tipo de dato	temario	descripcion
PK	id_libro	int		identificador del libro
	titulo	varchar	100	titulo del libro
	año	date		año publicacion del libro
FK	id_editorial	int		identificador del editorial

nombre: Lector

descripcion: almacena informacion de los lectores

clave	campo	tipo de dato	temario	descripcion
PK	id_lector	int		identificador del lector
	nombre	varchar	100	nombre del lector
	fecha_registro	date		fecha registro del lector



Create database odontologia

use odontologia;

Create table Paciente

Id_Paciente int primary key,

Nombre varchar (100),

Cedula varchar (100)

;

Create table Odontologos

Id_Odontologo int primary key,

Nombre varchar (100)

;

Create table Consultorios

Id_Consultorio int primary key,

Numero int,

Piso int

Create table citas

id_cita int primary key,

fecha date,

hora Time,

id_paciente int foreign key references paciente (id_Paciente),

id_odontologo int foreign key references odontologo (id_odontologo),

id_consultorio int foreign key references consultorio (id_consultorio)

);

Create table tratamiento

id_tratamiento int Primary Key,

nombre varchar (100)

);

Create table cita-tratamiento

id_cita int foreign key references cita (id_cita),

id_tratamiento int foreign key references tratamiento (id_tratamiento),

);

Create table odontologo_especialidad

id_odontologo int foreign key references odontologo (id_odontologo),

especialidad varchar (100)

);

insert into paciente values (1, 'Juan Perez', '101500010'),

insert into paciente values (2, 'Maria rico', '10001980'),

insert into odontologo values (1, 'Dra. Natalia Gomez'),

insert into consultorio values (1, 101, 1),

insert into citas values (1, '2023-07-10', '08:30:00', 1, 1, 1),

insert into tratamiento values (1, 'Limpieza dental'),

insert into cita-tratamiento values (1, 1),

insert into odontologo_especialidad values (1, 'ortodoncia')

select * from Pacientes;

select nombre from Paciente where cedula = '10250000';

select nombre from Paciente where id_Paciente in (

select id_Paciente from citas)

1)j

select * from Odontologo;

select nombre from odontologo where id_Odontologo = 1; j

select * from consultorios;

select nombre from consultorio where Piso = 1;

select nombre from consultorio where id_consultorio in (

select id_consultorio from citas)

7)j

select * from citas;

select Fecha from citas where id_Paciente = 1;

select hora from citas where id_Odontologo = 1;

select * from tratamientos;

select nombre from tratamiento where nombre like '%Limpieza';

select * from cita_tratamiento;

select * from odontologo_Especialidad;

select id_cita from cita_tratamiento where id_tratamiento = 1;

Paciente),
Odontologo),
(id_consultorio)

1)j

tratamiento)

odontologo),

1)j.

1)j

nombre: Cita
descripcion: clínica
Clave : Citas
PK : id_citas
FK : id_paciente
FK : id_odontologo
FK : id_consultorio
FK : id_tratamiento

nombre: Paciente
descripcion: paciente
Clave : Paciente
PK : id_paciente
FK : id_consultorio

nombre: Tratamiento
descripcion: procedimiento
Clave : Tratamiento
PK : id_tratamiento
FK : id_cita

Select especialidad from odontologo- especialidad

where id-odontologo IN

select id-odontologo from cita where fecha = '2015-07-10'

)j

select nombre from tratamiento

where id-tratamiento in

select id-tratamiento from cita-tratamiento

where id-cita in

select id-cita from cita

where id-paciente = c

select id-paciente from paciente where nombre = 'Juan Perez'

)

)

)j

Select c.fecha, c.hora, p.nombre As paciente, o.nombre As odontologo
From citas c

Join paciente p on c.id-paciente = p.Paciente

Join odontologos o on c.id-odontologo = o.id-odontologo;

Select o.nombre , oe.especialidad

From odontologo o

Join odontologo- especialidad on o.id-odontologo=oe.id-odontologo;

Delete from citas where id-citas;

Drop table citas;

nombre: citas

descripción: almacena información de cita

Clave	Campo	Tipo de dato	Largo	Descripción
PK	id-cita	int		identificador de cita
	fecha	date		fecha de la cita
Fk	id-paciente	int		identificador del paciente
Fk	id-odontologo	int		identificador de odontólogo
Fk	id-consultorio	int		identificador de consultorio

nombre: paciente

descripción: almacena información del paciente

Clave	Campo	Tipo de dato	Largo	Descripción
PK	id-paciente	int		identificador de paciente
	nombre	varchar	100	nombre del paciente
	cedula	varchar	100	cedula del paciente

nombre: consultorio

descripción: almacena información de consultorio

Clave	Campo	Tipo de dato	Largo	Descripción
PK	id-consultorio	int		identificador del consultorio
	numero	int		numero del consultorio
	piso	int		piso del consultorio

Select especialidad from odontologo_especialidad
 Where id_odontologo = 1
 Select id_tratamiento from tratamiento

nombre: odontologo				
descripcion: almacena informacion del odontologo				
clave	campo	tipo de dato	largo	descripcion
PK	id_odontologo	int	100	identificador del odontologo

nombre: odontologo
 descripcion: almacena informacion del odontologo
 clave: id_odontologo tipo de dato: int largo: 100
 nombre: nombre tipo: varchar

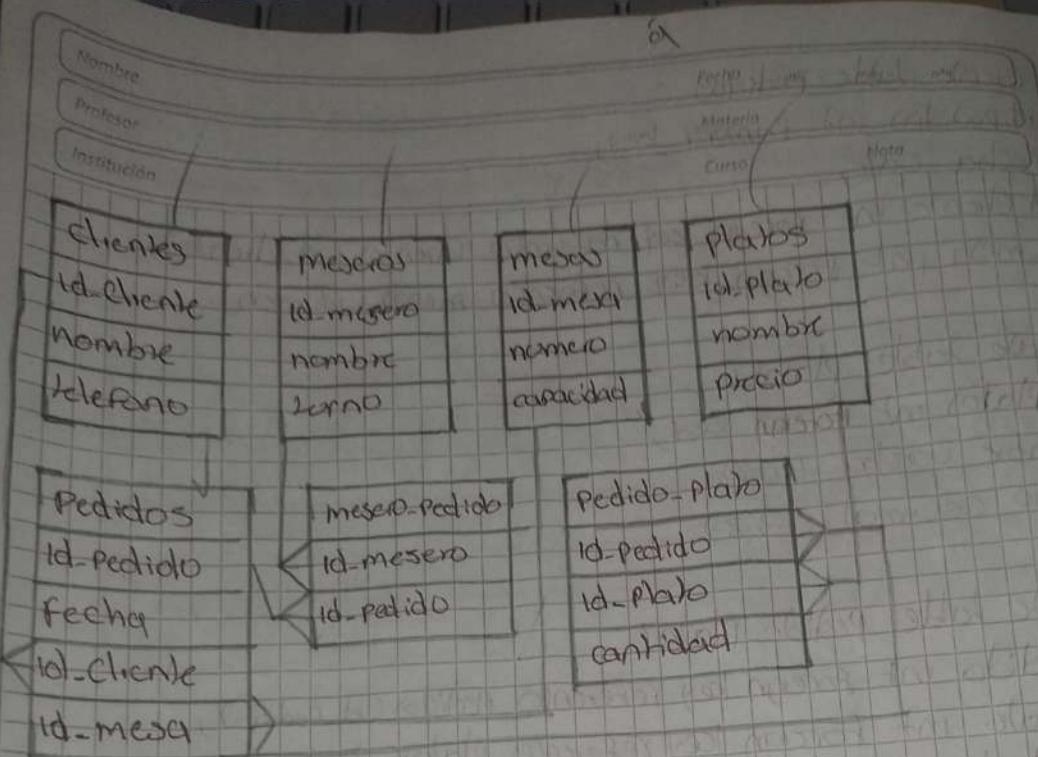
nombre: tratamiento				
descripcion: almacena informacion del tratamiento				
clave	campo	tipo de dato	largo	descripcion
PK	id_tratamiento	int	100	identificador de tratamiento

nombre: cita_tratamiento				
descripcion: tabla pivote				
clave	campo	tipo de dato	largo	descripcion
PK	id_cita	int		identificador de cita

nombre: odontologo_especialidad

descripcion: tabla pivote

clave	campo	tipo de dato	largo	descripcion
FK	id_odontologo	int		identificador de odontologo
FK	id_especialidad	int		identificador de especialidad



Create database restaurante;

use restaurante;

Create table Clientes(

id-cliente int primary key,

nombre varchar (100),

telefono varchar (100)

);

Create table Meseros(

id-meseros int primary key,

nombre varchar (100),

turno varchar (100)

);

Create table Mesas(

id-mesa int primary key,

numero int,

capacidad int



Create table Pedidos C
id-pedidos int primary key,

Fecha date,
id-cliente int foreign key references Clientes (id-cliente),
id-mesa int foreign key references mesas (id-mesa)

);

Create Table platos C
id-Platos int primary key,
Nombre varchar (100),
Precio decimal (10, 2)

);

Create Table Pedido-Plato C
id-Pedido int foreign key references Pedidos (id-Pedido),
id-Plato int foreign key references platos (id-Plato),
Cantidad int

);

Create Table mesero-Pedido C

id-mesero int foreign key references meseros (id-mesero),
id-Pedido int foreign key references Pedidos (id-Pedido)

);

Insert into Clientes values (1, 'Andrea', '30001');

Insert into meseros values (1, 'Luis', 'noche');

Insert into mesas values (1, 12, 4);

Insert into Pedidos values ('2023-07-09', 1, 1);

Insert into platos values (1, 'Pasta', 23000);

Insert into Pedido-Plato values (1, 1, 2);

Insert into mesero-Pedido values (1, 1);

select * from Clientes;
 select nombre from Clientes where telefono Like '7-3007';
 select nombre from Cliente where id-Cliente in(
 select id-Cliente from Pedidos)
);
 select * from meseros;
 select nombre from meseros where id-mesero in
 select id-mesero from mesero-Pedido
);
 select nombre from meseros where turno='noche';
 select * from mesas;
 select numero from mesas where id-mesa in(
 select id-mesa from Pedido)
);
 select * from Pedidos;
 select fecha from Pedido where id-Cliente = 1;
 select * from Platos;
 select nombre from Platos where id-Plato in(
 select id-Plato from Pedido-Plato
);
 select * from Pedido-Plato
 select id-Pedido from mesero-Pedido where id-mesero = 1;

nombre_Pedidos	
descripcion	dinner
clave	CAMPO
PK	id-Pedido
	fecha
FK	id-cliente
FK	id-mesero

nombre-Clm	
descripcion	
clave	C
PK	id

nom
des
clav
PK

Select nombre from Clientes

Where id_Cliente in (

Select id_Cliente from pedidos

i;

Select nombre from plato

Where id_Plato in (

Select id_Plato from Pedido_Plato

Where id_Plato in (

Select id_Pedido from Pedidos

Where id_Cliente = (

Select id_Cliente from Cliente where nombre = 'Andrea'.

i

)

);

Select nombre from meseros

Where id_meseros in (

Select id_mesero from mesero_pedido

Where id_Pedido in (

Select id_Pedido from Pedido where Fecha = '2023-07-01'

i

);

Select me.mesero As mesero , Pe.id_Pedido

From mesero_pedido Pe

Join meseros me On Pe.id_mesero = me.id_mesero;

Delete from Pedidos where id_Pedido = 1;

Drop table Pedidos;

Select nombre from Clientes

9

nombre: Pedidos

descripcion: almacena informacion de pedidos

clave	campo	tipo de dato	tamaño	descripcion
PK	id-pedido	int		identificador de pedido
	fecha	date		fecha de pedido
FK	id-cliente	int		identificador de cliente
FK	id-mesero	int		identificador de mesero

nombre: Clientes

descripcion: almacena informacion de Clientes

clave	campo	tipo de dato	tamaño	descripcion
PK	id-cliente	int		identificador del cliente
	nombre	varchar	100	nombre del cliente
	telefono	varchar	100	telefono del cliente

nombre: mesero

descripcion: almacena informacion de meseros

clave	campo	tipo de dato	tamaño	descripcion
PK	id-mesero	int		identificador de mesero
	nombre	varchar	100	nombre del mesero
	turno	varchar	100	turno del mesero

Select nombre from Clientes
 Where id_cliente in
 (select id_cliente from Pedidos)

Nombre: mesas

Descripción: Almacena información de mesas

clave	campo	tipo de dato	tamaño	descripción
PK	id-mesa	int		Identificador de mesa
	numero	int		número de la mesa
	capacidad	int		capacidad de la mesa

Nombre: platos

Descripción: Almacena información de platos

clave	campo	tipo de dato	tamaño	descripción
PK	id-plato	int		Identificador del Plato
	nombre	varchar	100	nombre del Plato

Nombre: mesero-Pedido

Descripción: Tabla PIVOTE

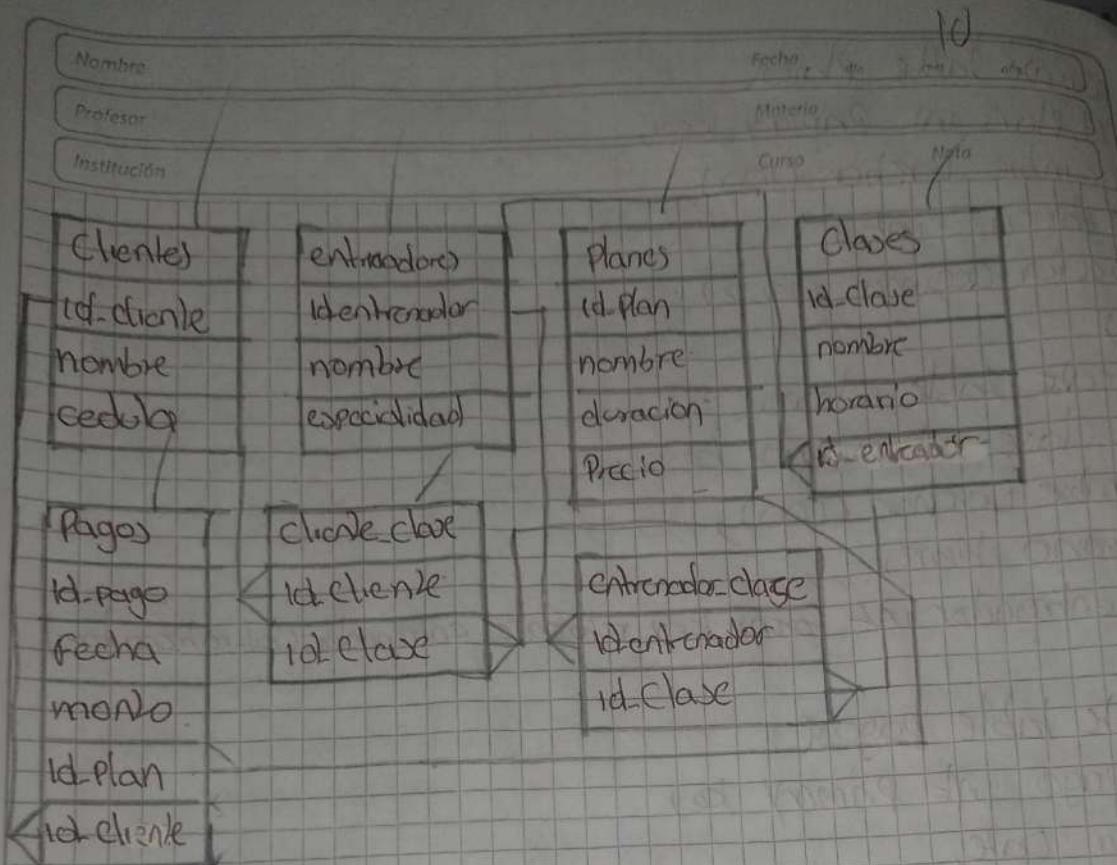
clave	campo	tipo de dato	tamaño	descripción
FK	id-mesero	int		Identificador de mesero
FK	id-pedido	int		Identificador de Pedido

Nombre: Pedido_Plato

Descripción: Tabla PIVOTE

clave	campo	tipo de dato	tamaño	descripción
FK	id-pedido	int		Identificador de Pedido
FK	id-plato	int		Identificador de Plato

10



Create database gimnasio;

use gimnasio;

Create table Clientes (

id_cliente int Primary Key,

nombre varchar (100),

cedula varchar (100)

);

Create table Entrenadores (

id_entrenador int Primary Key,

nombre varchar (100),

especialidad varchar (100)

);

Create Table planes
Id_Plan int primary key,
Nombre varchar(100),
Duracion int,
Precio decimal(10,2)

);

Create Table Clases
Id_Clase int primary key,
Nombre varchar(100),
Horario time,

Id_entrenador int foreign key references Entrenador(id_entrenador)

);

Create Table Pagos

Id_Pago int primary key

Fecha date,

Monto decimal(10,2),

Id_Cliente int foreign key references Cliente(id_Cliente),

Id_Plan int foreign key references Planes(id_Plan)

);

Create Table Cliente_Clase

Id_Cliente int foreign key references Cliente(id_Cliente),

Id_Clase int foreign key references Clase(id_Clase)

);

Create Table Entrenador_Clase

Id_entrenador int foreign key references Entrenador(id_entrenador),

Id_Clase int foreign key references Clase(id_Clase);

);

insert into Clientes values (1, 'Carlos Ruiz', '12345');
insert into Entrenador values (1, 'Laura', 'cardio');
insert into Planes values (1, 'Plan General', 1, 50.000);
insert into Clases values (1, 'Spinning', '07:00:00', 1);
insert into Pagos values (1, '2023-07-04', 50000, 1, 1);
insert into Cliente_Clase values (1, 1);
insert into Entrenador_Clase values (1, 1);

select * from Clientes;

select nombre from Clientes where cedula = '12345';

select * from Entrenadores;

select nombre from Entrenadores where id_entrenador in (

select id_entrenador from Entrenador_Clase

);

select * from Planes;

select nombre from Planes where id_plan in (

select id_plan from Pagos

);

select * from Clases;

select nombre from Clases where horario < '12:00:00';

select * from Pagos;

select monto from Pagos where id_cliente = 1;

select * from Cliente_Clase;

select id_cliente from Cliente_Clase where id_clase = 1;

select id_cliente from Cliente_Clase where id_clase = 1;

Select * from entrenador_clase;

Select id_entrenador from entrenador_clase where clase = 1;

Select nombre from Cliente;

Where id_cliente in C;

Select id_cliente from Pago;

);

Select nombre from entrenadores;

Where id_entrenador in C;

Select id_entrenador from Clases where horario < '10:00:00';

);

Select P.fecha, P.monto, C.nombre As Cliente, Pl.nombre As Plan

From Pagos P

Join Cliente C on P.id_cliente = C.id_cliente

Join Planes Pl on P.id_plan = Pl.id_plan;

Select C.nombre As clase, E.nombre As entrenador

From Clases C;

Join entrenadores E on C.id_entrenador = E.id_entrenador;

Select C.nombre As cliente, Cl.nombre As clase

From Cliente_Clase CC

Join Cliente C on CC.id_cliente = C.id_cliente

Join Clases Cl on CC.id_clase = Cl.id_clase;

Delete From Pagos where id_Pago = 1;

Drop Table Pagos;

Select * from entrenador_Clase;

Select id_entrenador From entrenador_Clase where clase = 1;

nombre: cliente				
descripcion: almacena informacion de clientes				
clave	campo	tipo de dato	tamaño	descripcion
PK	id-cliente	int		identificador de cliente

nombre: entrenador				
descripcion: almacena informacion de entrenador				
clave	campo	tipo de dato	tamaño	descripcion
PK	id_entrenador	int		identificador de entrenador

nombre: Cliente_Clave				
descripcion: tabla Pilote				
clave	campo	tipo de dato	tamaño	descripcion
FK	id-cliente	int		identificador de cliente
FK	id-clave	int		identificador de clase

nombre: entrenador_Clave				
descripcion: tabla Pilote				
clave	campo	tipo de dato	tamaño	descripcion
FK	id_entrenador	int		identificador de entrenador
FK	id_clave	int		identificador de clase

Select * from entenador.clave;

Select id_entenador from entenador.clave where clave = 1;

10

nombre : Pago)

descripcion: almacena informacion de pago

clave	campo	tipo de dato	tamano	descripcion
PK	id_pago	int		identificador de pago
	fecha	date		fecha de pago
	monto	int		monto de pago
FK	id_plan	int		identificador de plan
FK	id_cliente	int		identificador de cliente

nombre : Clase)

descripcion: almacena informacion de clase

clave	campo	tipo de dato	tamano	descripcion
PK	id_clase	int		identificador de clase
	nombre	varchar	100	nombre de clase
	horario	time		horario de clase
FK	id_entenador	int		identificador de clase

nombre : Plan)

descripcion: almacena informacion de plan

clave	campo	tipo de dato	tamano	descripcion
PK	id_plan	int		identificador del plan
	nombre	varchar	100	nombre del plan
	duration	time		duration del plan
	precio	decimal	10, 2	precio del plan