

Create database `votes` - `spoolades`
use `votes` - `spoolades`;

Self talk stories of self-control may also increase positive self-talk (Wade et al., 2005).

Create table novel
id - nove in primary key auto increment
nombe branch (C50)

>Create table migration
id - migration
name - varchar(250)
label - text
foreign key (id) references have(id)

Create table planetac
10 planets in memory for each movement

Creatable table repeated
id repeat, in primary key auto-increment

id mission int remember wörter (Coa)
Fachdaten ?

Foreign key (id mission) references mission (Create table tryplace(id mission))

10-osteoarthritis in Cid patients

French key (old mission), (old costalard)

• *Yesterdays offer today mission plan offer*

negative feedback mission - pioneer (i)
id_mision, id_pioneer
id_planeta, id_planeta_id
binary key, Cd_mision (Cd_planeta)
foreign key (Cd_mision) references mision Cd_mision
foreign key Cd_planeta references planeta (id_planeta))

myself into as many channels (Chambres) as best

(Hoff Whitney)
(Offer Serey)
(Plan)

July 19, 1944 - 100 miles S.E. of Laramie, Wyo.

Ckessel Run', 31'2025-05-09D';
insert into planet (nombre, tipo) values

'Earth', 'terrestre');

insert into planet (nombre, tipo)

'Mars', 'terrestre');

insert into planet (nombre, tipo)

'Jupiter', 'gasoso');

insert into planet (nombre, tipo)

'Saturn', 'gasoso');

insert into planet (nombre, tipo)

'Uranus', 'gasoso');

insert into planet (nombre, tipo)

'Neptune', 'gasoso');

insert into planet (nombre, tipo)

'Mercury', 'terrestre');

insert into planet (nombre, tipo)

'Venus', 'terrestre');

insert into planet (nombre, tipo)

'Pluto', 'terrestre');

insert into planet (nombre, tipo)

'Eris', 'terrestre');

insert into planet (nombre, tipo)

'Haumea', 'terrestre');

insert into planet (nombre, tipo)

'Ceres', 'terrestre');

insert into planet (nombre, tipo)

'Ganymede', 'satelite');

insert into planet (nombre, tipo)

'Europa', 'satelite');

insert into planet (nombre, tipo)

'Callisto', 'satelite');

insert into planet (nombre, tipo)

'Titán', 'satelite');

insert into planet (nombre, tipo)

'Iapetus', 'satelite');

insert into planet (nombre, tipo)

'Hyperion', 'satelite');

insert into planet (nombre, tipo)

'Phoebe', 'satelite');

insert into planet (nombre, tipo)

'Prometeo', 'satelite');

insert into planet (nombre, tipo)

'Atlas', 'satelite');

insert into planet (nombre, tipo)

'Janus', 'satelite');

insert into planet (nombre, tipo)

'Epimeteo', 'satelite');

insert into planet (nombre, tipo)

'Telesto', 'satelite');

insert into planet (nombre, tipo)

'Helene', 'satelite');

insert into planet (nombre, tipo)

'Puck', 'satelite');

insert into reporte (id_mision, resumen, fecha) values
(1, 'reparación de muestras', '2030-08-07'),
(2, 'encuentro hasta con teroma los', '2122-12-10'),
(3, 'completo esp/2 pañecos', '2025-09-05');

insert into tripulacion (id_mision, id_astronauta)

(1, 1), (1, 2),

(2, 2), (2, 3),

(3, 2), (3, 3),

(1, 1), (2, 2), (3, 3), (3, 4);

- Eliminaciones

-1) Quitar toda tripulación de la misión 2

delete from misión_planeta

where id_misión = 2;

-1) tripulación por misión (inner join)

select m.nombre as misión,

a.nombre as astronauta

from tripulacion t

join misión m on t.id_misión = m.id_misión

join astronauta a on t.id_astronauta = a.id_astronauta

-2) misiones y sus repartos

select m.nombre as misión,

r.resumen

from misión m

left join reparto r on m.id_misión = r.id_misión;

-3) planetas y misiones

select p.nombre as planeta,

m.nombre as misión

from planeta p

join misión_planeta mp on mp.id_misión = m.id_misión;

right join misión m on mp.id_misión = m.id_misión;

right join planeta p on mp.id_planeta = p.id_planeta;

- q. cantidad de misiones por race
 - select nombre as race
 - count(m.id_mision) as total_misiones
- from race n
 - left join mision m on mid_race = m.id_race
 - group by n.nombre;
- 5. tripulantes por mision (sum nro (cont))
 - select sum.nro as mision
 - count.id_mision as tripulantes
- from mision m
 - join tripulacion t on m.id_mision = t.id_mision
 - group by m.nombre;
- 6. promedio de tripulantes por mision (avg sobre subconsulta)
 - select avg(cnt) as promedio
 - from tripulacion t
 - join mision m
 - group by m.nombre;
- 7. mision con mas tripulantes
 - create subquery
 - select promedio, count from(
 - select m.nombre, count(*) as cont
 - from mision m
 - join tripulacion t
 - group by m.nombre;
- from temp
 - where cont = (select max(cont) from(
 - select nombre, count(*) as cont
 - from mision m
 - join tripulacion t
 - group by m.nombre;
)
- 8. mision con menos planetas visitados (min con left join)
 - select nombre, cont from(
 - select m.nombre, count(*) as cont
 - from mision m
 - left join planetas p
 - group by m.id_mision = p.id_mision
 - group by p.nombre;
)
 - where cont = (select min(cont) from(
 - select nombre, count(*) as cont
 - from mision m
 - left join planetas p
 - group by m.id_mision
 - group by p.nombre;
)

Create table base_gestion_proyectos;
Vx - gestion - proyectos;

Create table usuarios (id_usuario int auto-increment primary key,
nombre varchar(50) not null);

2;

Create table proyecto (id_proyecto int auto-increment primary key,
nombre varchar(50) not null);

3;

Create table equipo (id_equipo int auto-increment primary key,
rel_proyecto int not null);

4;

Create table factura (id_factura int auto-increment primary key,
id_proyecto int not null null,
fecha_voucher (100) not null,
fecha_entrega (30) not null);

5;

foreign key (id_proyecto) references proyecto;

6;

foreign key (id_proyecto) references proyecto (cascada);

7;

Create table usuario_remps (id_remps int not null auto-increment primary key,
id_factura int not null null,
fecha_desemb (100) not null);

8;

foreign key (id_factura) references factura (id_factura) on
Delete cascade;

9;

foreign key (id_usuario) references usuarios (id_usuario) on
delete cascade;

10;

Create table equipo_usuarios (id_equipo int not null,
id_usuario int not null);

11;

primary key (id_equipo id_usuario);

12;

foreign key (id_equipo) references equipo (id_equipo);

13;

foreign key (id_usuario) references usuarios (id_usuario);

14;

on delete cascade;

15;

Create table factura_usuarios (id_factura int not null,
id_usuario int not null);

16;

primary key (id_factura id_usuario);

17;

foreign key (id_factura) references factura (id_factura);

18;

foreign key (id_usuario) references usuarios (id_usuario);

19;

on delete cascade;

20;

Create table base_gestion_usuarios (id_usuario int auto-increment primary key,
nombre varchar(50) not null);

21;

Create table factura_usuarios (id_factura int auto-increment primary key,
id_usuario int not null);

22;

foreign key (id_usuario) references base_gestion_usuarios (id_usuario);

23;

on delete cascade;

24;

insert into vmo (nombre) values
('ana'),
('luis'),
('mario');

insert into projects (Nombre) values
(('Alta')
(('Baja'));

insert into equipo (Nombre) values
(('yo'),
(('azul')),
(('verde'));

insert into tarea (id_proyecto, tipo) values
({1, 'registrar'),
({1, 'desarrollar'),
({2, 'desarrollar'),
({3, 'probar'),
({3, 'soporte}),

insert into regishto. numero (tarea, claves, horas) values

{1, 1, 8.00},
{1, 2, 6.50},
{2, 2, 3.15},
{3, 2, 10.00},
{4, 2, 5.00},
{5, 3, 7.75},

insert into equipo (claves) values

{1, 1},
{1, 2},
{2, 1},
{3, 2},
{3, 3},
{3, 1},
{5, 2},
{3, 2},
{5, 3},

insert into tarea - clave (id_tarea, claves) values

{1, 1},
{1, 2},
{2, 1},
{3, 2},
{3, 3},

insert from regishto. numero where id_registro = 1 into tarea - clave (tarea = 3 and id_usuario = 3) and id_proyecto = 1;

update usuario set nombre = 'ana' where id_usuario = 1;

update proyecto set nombre = 'alta' where id_proyecto = 1;

set proyecto = 1 en proyecto where id_tarea = 3;

update regishto. numero set horas = 8.50 when id_registro = 1;

- creando de horas registradas
Select v.nombre as nombre, r.t_id as horas
from registro_tarea r t
join persona p on r.t_id = p.id_usuario
join tarea t on r.t_id = t.id_tarea = p.id_tarea;

Select t.titulo as tarea, t.v.id_usuario
from tarea t
left join persona p on t.id_tarea = p.id_tarea;

Select e.nombre as equipo, v.nombre as miembros
from equipo e
join equipo v on e.id_equipo = v.id_equipo
join grupo g on v.id_usuario = v.id_usuario;

- contar de horas por proyecto, v.nombre as proyecto
Select p.nombre as proyecto, count(t.id_tarea) as total_hours
from proyecto p
join tarea t on p.id_proyecto = t.id_tarea
join registro_tarea rt on t.id_tarea = rt.id_tarea
group by p.nombre;

- promedio de horas por tarea, avg(t.horas) as horas_promedio
Select et.titulo as tarea, avg(t.horas) as max_horas
from tarea t
join registro_tarea rt on t.id_tarea = rt.id_tarea
group by t.titulo;

- buscar con maximo de horas as max-horas
Select p.titulo as max_hours
from persona p
join registro_tarea rt on p.id_persona = rt.id_tarea
group by p.titulo
order by max_hours desc;

- equipos con mas miembros Count(COUNT.id_usuario) as miembros
Select e.nombre, COUNT(COUNT.id_usuario) as miembros
from equipo e
join equipo v on e.id_equipo = v.id_usuario
group by e.nombre
having miembros (contar) from (count(id_usuario)) as contaclos
Select max(count(id_usuario)) as contaclos
from equipo e
group by e.nombre
group by id_equipo

group by sub

- horas con mas horas sum(t.horas) as total_hours
Select v.nombre, sum(t.horas) as total_hours
from usuario v
join registro_tarea rt on v.id_usuario = rt.id_usuario
join persona p on v.id_usuario = p.id_usuario
group by v.id_usuario
having sum(t.horas) from (

Select sum(aug.total) from (

Nombre	Fecha
Profesor	dia
Institución	mes
	año
	Materia
	Curso
	Nota

Create database panaderiaDB;
use panaderia db;

```

Create table cliente(
    id_cliente int auto_increment primary key,
    nombre varchar(100),
    telefono varchar(20),
    direccion varchar(150),
);

Create table empleado(
    id_empleado int auto_increment primary key,
    nombre varchar(100),
    cargo varchar(50),
);

Create table producto(
    id_producto int auto_increment primary key,
    nombre varchar(100),
    precio decimal(10,2),
    descripcion text,
);

Create table receta(
    id_receta int auto_increment primary key,
    id_producto int,
    instrucciones text,
    Foreign key (id_producto) references producto(id_producto)
);

Create table ingrediente(
    id_ingrediente int auto_increment primary key,
    nombre varchar(100),
    unidad_media varchar(20)
);

Create table detalle_receta(
    id_detalle int auto_increment primary key,
    id_receta int,
    id_ingrediente int,
    cantidad decimal(10,2),
    Foreign key (id_receta) references receta(id_receta),
    Foreign key (id_ingrediente) references ingrediente(id_ingrediente)
);

Create table venta(
    id_venta int auto_incremente primary key,
    id_cliente int,
    id_empleado int,
    id_producto int,
    fecha datetime,
    cantidad int,
    total decimal(10,2),
    Foreign key (id_cliente) references cliente(id_cliente),
    Foreign key (id_empleado) references empleado(id_empleado),
    Foreign key (id_producto) references producto(id_producto)
);

Insert into Cliente (nombre, telefono, direccion) values
('Carla Pérez', '3101234567', 'calle 10 #5-21'),
('Carlos Fernández', '3111234567', 'Carrera 12 #8-34');

```

Insert into empleado (Nombre, cargo) values ('Jesús Conde', 'vendedor');
Insert into producto (Nombre, precio, descripción) values ('Pan Francés', 1600.00, 'pan tradicional (crujiente)');
Insert into ingrediente (Nombre, UnidadMedida) values ('harina', 'gramos'), ('levadura', 'gramos'), ('agua', 'mililitros'), ('sá', 'gramos'), ('manteca', 'gramos');
Insert into receta (id_producto, instrucciones) values (1, 'mezclar los ingredientes y hornear 20 minutos.'), (2, 'amasar, reposar y hornear el croissant');
Insert into detalle_receta (id_receta, id_ingrediente, cantidad) values (1, 1, 500), -- harina
(1, 2, 10), -- levadura
(1, 3, 300), -- agua
(1, 4, 5), -- sal
(2, 1, 400), -- harina
(2, 5, 100); -- manteca
Insert into venta (id_cliente, id_empleado, id_producto, Fecha, cantidad, total) values (1, 1, 1, now(), 3, 4800.00), (2, 2, 2, now(), 2, 500.00);

-- Eliminar una venta específica

delete from venta
where v.id_venta=2;

-- Eliminar el producto 'croissant' de la base de datos

delete from detalle_receta

where id_receta = (Select r.id_receta from receta as r where r.id_producto=2);

delete from receta

where id_producto=2;

delete from venta

where v.id_producto=2;

delete from producto

where p.id_producto=2;

-- Cambiar el nombre del empleado con id 1

update empleado

set nombre='Jesús Conde'

where e.id_empleado=1;

-- Cambiar el precio del pan francés

update producto as p

set p.precio=1600.00

where p.id_producto=1;

-- Cambiar el nombre de un cliente

update cliente as c

set c.nombre='Carlos Fernández'

where c.id_cliente=2;

-- Cambiar la cantidad vendida en una venta

update venta as v

set v.cantidad=5

where v.id_venta=1;

Select

c.nombre as cliente,
p.nombre as producto,
e.nombre as empleado,
u.cantidad
v.fecha,
u.fecha

From venta as v
inner join cliente as c on v.id_cliente = c.id_cliente
inner join producto as p on v.id_producto = p.id_producto
inner join empleado as e on v.id_empleado = e.id_empleado;

Select

p.nombre as producto,
i.nombre as ingrediente,
dr.cantidad

From receta as r
left join detalle_receta as dr on r.id_receta = dr.id_receta
left join ingrediente as i on dr.id_ingrediente = i.id_ingrediente
inner join producto as p on r.id_producto = p.id_producto;

Select

i.nombre as ingrediente,
dr.cantidad,
r.id_receta

From detalle_receta as dr
right join ingrediente as i on dr.id_ingrediente = i.id_ingrediente
left join receta as r on dr.id_receta = r.id_receta;

Select

p.nombre as producto,
sum(v.total) as total_vendido

From venta as v
inner join producto as p on v.id_producto = p.id_producto
group by p.nombre;

Select

c.nombre as cliente,
count(v.id_venta) as cantidad_ventas

From venta as v
inner join cliente as c on v.id_cliente = c.id_cliente
group by c.nombre;

Select

c.nombre as cliente,
avg(v.total) as promedio_gasto

From venta as v
inner join cliente as c on v.id_cliente = c.id_cliente
group by c.nombre;

Select

p.nombre as producto,
p.precio

From producto as p
where p.precio = (Select max(precio) From producto);

Select

c.nombre as cliente

From cliente as c
Where c.id_cliente in
select v.id_cliente
From venta as v

Create database productora_musical;
use productora_musical;

Create table artista (
id_artista int auto-increment primary key,
nombre varchar(100),
genero varchar(50),
nacionalidad varchar(50)
);

Create table album (
id_album int auto-increment primary key,
titulo varchar(100),
ano int,
duracion_total time
);

Create table Cancion (
id_cancion int auto-increment primary key,
titulo varchar(100),
duracion time,
id_album int,
foreign key (id_album) references album (id_album)
);

Create table productor (
id_productor int auto-increment primary key,
nombre varchar(100),
experiencia_anos int
);

Create table estudio (
id_estudio int auto-increment primary key,
nombre varchar(100),
ciudad varchar(50)
);

Create table artista_album (
id_relacion int auto-increment primary key,
id_artista int,
id_album int,
foreign key (id_artista) references artista (id_artista),
foreign key (id_album) references album (id_album)
);

Create table grabacion (
id_grabacion int auto-increment primary key,
id_cancion int,
id_estudio int,
id_productor int,
fecha_grabacion date,
foreign key (id_cancion) references cancion (id_cancion),
foreign key (id_estudio) references estudio (id_estudio),
foreign key (id_productor) references productor (id_productor)
);

-- inserts

insert into artista (nombre, genero, nacionalidad) values
('shakira', 'pop', 'colombiana'),
('bad bunny', 'reggaeton', 'puertorriqueño'),
('juanes', 'rock', 'colombiano');

insert into album (titulo, año, duracion_total) values
('el cibado', 2017, '00:40:00'),
('yohiamal', 2020, '00:50:00'),
('origen', 2021, '00:45:00');

insert into cancion (id_cancion, id_album) values
('charafe', '00:03:30'),
('yo perdi sola', '00:03:15'),
('volverte aver', '00:04:00');

insert into productor (nombre, experiencia_anos) values
('andres castro', 15),
('tainy', 10),
('sebastian krys', 20);

insert into estadio (nombre, ciudad) values
('estadio alpha', 'bogota'),
('la base studio', 'dcm juan'),
('circle house', 'miami');

insert into artista_album (id_artista, id_album) values
(1, 1),
(2, 2),
(3, 3);

insert into grabacion (id_cancion, id_estadio, id_productor,
fecha_grabacion) values
(1, 1, '2016-08-10'),
(2, 2, '2019-11-25'),
(3, 3, '2020-05-12');

delete from grabacion where id_cancion = 2;

delete from cancion where id_cancion = 2;

update artista

set genero = 'latin pop'
where id_artista = 1;

update cancion

set duracion = '00:03:45'
where id_cancion = 1;

update album

set duracion_total = '00:42:00'
where id_album = 1;

update productor

set experiencia_anos = 18
where id_productor = 1;

-- inner join

select

c. titulo as Cancion,

a. titulo as album,

ar. nombre as artista,

p. nombre as productor,

e. nombre as estudio,

g. fecha_grabacion

from grabaciones as g

inner join Cancion as c on g. id_cancion = c.id_cancion

inner join album as a on c.id_album = a.id_album

inner join artista_album as aa on a.id_album = aa.id_album

inner join artista as ar on aa.id_artista = ar.id_artista

inner join productor as p on g.id_productor = p.id_productor

inner join estudio as e on g.id_estudio = e.id_estudio;

-- left join

select

a. titulo as album,

ar. nombre as artista

from album as a

left join artista_album as aa on a.id_album = aa.id_album

left join artista as ar on aa.id_artista = ar.id_artista;

Select

ar. nombre as artista,

a. titulo as album

from artista as ar

right join artista_album as aa on ar.id_artista = aa.id_artista

left join album as a on aa.id_album = a.id_album;

-Count

select g. titulo as album, count(c.id_cancion) as fofo_canciones

from album as a

left join Cancion as c on a.id_album = c.id_album

group by a.titulo;

-Sum

select sec_to_time(sum(time_to_sec(c.duration))) as

duracion_foto

from canciones as c;

-- avg

select a.titulo as album, sec_to_time(avg(time_to_sec(dur)))

as promedio

from canciones as c

inner join album as a on c.id_album = a.id_album

group by a.titulo;

-max

select t.titulo, duracion

from canciones as t

where duracion = (select max(duracion) from canciones)

-Count

select count(cld.nacionalidad) as nacionalidades_diferentes

from artista;

+ Subconsulta en in

Select g.id AS album
From album AS g
Where a.id_album IN (

Select id_album
From cancion
Where duracion > '00:03:30'
)

- subconsulta anidada

Select nombre
From productor
Where id_productor IN (

Select id_productor

From grabacion

Where id_cancion

From cancion

Where duracion > '00:03:00'

)

- having

Select p.nombre AS productor, count(g.id_grabacion) AS total_canciones
From grabacion AS g
Inner Join productor AS p ON g.id_productor = p.id_productor
Group By p.nombre
Having total_canciones > 1;

Nombre	Fecha
Profesor	dia mes año
Institución	
	Materia
	Curso
	Nota

Create database refugio_animales;
use refugio_animales;

Create table especie (
id_especie int auto-increment primary key,
nombre_especie varchar(50)
);

Create table animal (
id_animal int auto-increment primary key,
nombre varchar(50),
edad int,
Sexo enum ('m', 'f'),
id_especie int
Fecha_nacimiento date,
foreign key (id_especie) references especie (id_especie)
);

Create table Cuidador (int auto-increment primary key,
nombre varchar(100),
telefono varchar(20)
);

Create table adoptante (
id_adoptante int auto-increment primary key,
nombre varchar(100),
direccion varchar(150),
telefono varchar(20)
);

Create table vacuna (
id_vacuna int auto-increment primary key,
nombre varchar(100),
clase varchar(20)
);

Create table animal_Cuidador (
id_relacion int auto-increment primary key,
id_animal int,
id_cuidador int,
fecha_asignacion date,
foreign key (id_animal) references animal (id_animal),
foreign key (id_cuidador) references cuidador (id_cuidador)
);

Vacunación

Create table vacunacion (
id_vacunacion int auto-increment primary key,
id_animal int,
id_vacuna int,
fecha_aplicacion date,
foreign key (id_animal) references animal (id_animal),
foreign key (id_vacuna) references vacuna (id_vacuna)
);

insert into especie (nombre_especie) values

('Perro'),

('gato'),

('conejo');

insert into animal (nombre, edad, sexo, cd_especie, fecha_ingreso) values

('luna', 3, 'M', '2024-06-01'),

('max', 2, 'M', '2024-06-15'),

('nieve', 1, 'F', '2024-06-01');

insert into cardador (nombre, telefono) values

('carla torres', '3101112233'),

('Carlos mejia', '3112223344');

insert into veterinario (nombre, dirección, telefono) values

('luis perel', 'cra 10 # 120 - 72', '3000000000'),

('luis gomez', 'c1 # 78-90', '34111111');

insert into vacuna (nombre, dosis_valores)

('rabia', '1ml'),

('parvovirus', '2ml'),

('fleble felina', '5ml');

insert into animal_cardador (cd_animal, cd_cardador, fecha_asignacion) values

(1, 1, '2024-06-01'),

(2, 2, '2024-06-16'),

(3, 1, '2024-07-02');

insert into vacunacion (cd_animal, cd_vacuna, fecha_aplicacion) values

(1, 1, '2024-06-02'),

(2, 2, '2024-06-18'),

(3, 1, '2024-07-05');

update from vacunacion where cd_vacuna = 1;

update from animal where nombre = 'max';

update animal

set edad = 4
where nombre = 'luna';

update cardador

set telefono = '3000000000'
where nombre = 'carlos mejia';

update vacuna

set doses = '1.8ml'

where nombre = 'fleble felina';

select

a.nombre as animal,

e.nombre as especie us especie

a.edad

a.fecha_ingreso

from animal as a

inner join especie as e on a.cd_especie = e.cd_especie

- left join

Select

c. nombre as animal,
c. nombre as cuidador,
ac. fecha - argentina

from animal as a

left join animal - cuidador as ac
a.id_animal = ac.id_animal

left join Cuidador as C
a.id_animal = C.id_cuidador

- right join

Select

c. nombre as cuidador,
a. nombre as animal

from Cuidador as c

right join animal - cuidador as ac on C.id_cuidador = ac.id_animal

left join animal as a on ac.id_animal = a.id_animal

Select

-- Count

e. nombre - especie
Count (a.id_animal) as cantidad

from especie as e

left join animal as a on e.id_especie = a.id_especie

group by e.nombre - especie;

- sum

Select

Count(*) as total_julio

from animal

where month(Fecha_ingreso) = 7;

-- avg

Select

avg(edad) as edad_promedio

from animal;

-- max

Select

nombre, edad

from animal

where edad = select max(edad) from animal;

-- min

Select

nombre, edad

from animal;

where edad = (select min(edad) from animal);

-- distinct

Select count(distinct sexo) as sexos_distintos

from animal

- subconsulta conn

Select nombre

from animal

where id_animal in (

Select id_animal

from vacunacion

group by id_animal

having count(id_vacuna) > 1

)

- sub conserve and select

- select number
from animal
where let - animal > 10

select id - animal
from useraccess
where let - animal in (

select id - value
from useraccess
where class = '1m1'

)

;

- having

Select

symbol as Cudecolor
Crom (ac_id animal) as total - animal

Crom Cudecolor as C
Inner join Cudcolor - Cudcolor as ac on C.ac_id - Cudecolor - ac_id

Cudecolor by Crom
group by Crom
having total - animals > 1

Crear database teorías conspirativas;
use teorías conspirativas;

Create table teoriasC
id_teoria int primary key auto_increment,
nombre varchar(50)
);

Create table fuentesC
id_fuente int primary key auto_increment,
nombre varchar(50)
);

Create table eventosC
id_evento int primary key auto_increment,
nombre varchar(50),
fecha date
);

Create table actoresC
id_actor int primary key auto_increment,
nombre varchar(50)
);

Create table teoria_FuenteC
id_tf int primary key auto_increment,
id_teoria int,
id_Fuente int
Foreign key (id_teoria) references teorias (id_teoria),
Foreign key (id_Fuente) references fuentes (id_Fuente)
);

Create table teoria_EventoC
id_te int primary key auto_increment,
id_teoria int,
id_evento int
Foreign key (id_teoria) references teorias (id_teoria),
Foreign key (id_evento) references eventos (id_evento)
);

Create table actor_TeoricaC
id_at int primary key auto_increment,
id_actor int,
id_teoria int
Foreign key (id_actor) references actores (id_actor),
Foreign key (id_teoria) references teorias (id_teoria)
);

Insert into teorias (Nombre) values
('Área 51 secreto militar'),
('reptilianos gobernan'),
('tierra plana');

Insert into fuentes (Nombre) values
(libros),
(canal 41 odiado),
(áfrica/az2);

Insert into eventos (Nombre, Fecha) values
('Roswell 1947', '1947-07-08'),
('Colonización 1969', '1969-07-20'),
('caida origen 2019', '2019-12-01');

'Insert into actors nombre values
(‘david_lickie’),
(‘lenny_mccoy’),
(‘james_cobell’);

Insert into teoria_fuente(id_teoria,id_Fuente) values
(1,1),
(1,2),
(2,2),
(3,2),
(3,3).

Insert into actor_fuente(id_actor,id_fuente) values
(1,1),
(1,2),
(2,2),
(3,1),
(3,2),
(3,3).

Insert into teoria_evento(id_teoria,id_evento) values
(1,1),
(1,2),
(2,1),
(2,2),
(3,1),
(3,2),
(3,3).

Insert into actor_fuente(id_actor,id_fuente) values
(1,1),
(1,2),
(2,1),
(2,2),
(3,1),
(3,2),
(3,3).

- Clipping la fuente 00 confiable de 'Tero plana'
delete from fuente where id_F=4;

- Elimina el evento 'cada' origen 2099,
delete from teoria_evento where id_Evento = 4;

- delete from eventos where id_evento = 3;

- Update teorias set punto='reptiliano en el desierto' where id_teoria=2;

- Update actores set nombre='d.lickie' where id_acto=1;

- Select inner join: teorias con sus fuentes
select nombre as teorias, t.nombre as Fuente
from teorias, t;

Inner join Teoria_Fuente tf on tf.id_teoria = tf.id_Teoria
inner join fuentes t on tf.id_Fuente = t.id_Fuente;

- Select left join: teorias y eventos registrados
select nombre as teorias, e.nombre as eventos

from teorias, t
left join teoria_evento te on t.id_teoria=id_Teoria
left join eventos e on te.id_Evento=id_Evento;

- Select right join: actores
select nombre as actores, t.nombre as teorias

right join actores t on t.id_actor=id_actor
right join teorias q on q.id_teoria=id_teoria;

- Count: contar cantidad de fuentes por teoria
Select count(*) as cantidad
from fuentes
group by id_teoria

Join teoria_Fuente tf on t.id_teoria=id_Teoria
join fuentes f on tf.id_Fuente=id_Fuente

Group by id_teoria
having count(Fuentes)>1;

- Subconsulta: teoria con maximo de fuentes
select nombre, max(Fuentes) as max_fuentes
from fuentes
group by id_teoria

Subquery by id_teoria
having count(Fuentes)=max_fuentes

where nom_Fuentes = C
select max(score) from C
select count(*) as cnt
from teoria_Fuentes
group by id_teoria

;

;

```
Create database objetos_malditos;
use objetos_malditos;
Create table objetosC
id_objeto int primary key auto-increment,
nombre varchar(250),
señal varchar(250);
Create table curadorC
id_curador int primary key auto-increment,
nombre varchar(250);
Create table origenC
id_origen int primary key auto-increment,
ubicacion varchar(250),
fecha date;
Create table eventoC
id_evento int primary key auto-increment,
descripcion varchar(250),
fecha date;
```

```
Create table inventarioC
id_inventario int primary key auto-increment,
id_objeto int,
resumen varchar(250),
fecha_date,
foreign key (id_objeto) references objeto_C(id_objeto);
```

```
Create table eventosC
id_evento int primary key auto-increment,
id_objeto int,
id_evento int,
foreign key (id_objeto) references objeto_C(id_objeto),
foreign key (id_evento) references evento_C(id_evento);
```

```
Create table curador_eventosC
id_coe int primary key auto-increment,
id_curador int,
id_objeto int,
foreign key (id_curador) references curador_C(id_curador),
foreign key (id_objeto) references objeto_C(id_objeto);
```

```
-- Insertar datos
insert into objeto(nombre,señal)
values ('murciel', 'allo'),
('espere sorgiento', 'medio'),
('libro sellado', 'extremo');
insert into curador (nombre)
values ('dr raven'),
('prof black'),
('srta eclipse');
```

```
insert into origen (ubicacion, fecha) values  
('monasterio de carpa', '2020-05-01'),  
('mansion holloway', '2020-11-23'),  
('cripta oluboda', '1953-01-12');
```

```
insert into evento (observacion, fecha) values  
('objeto levando', '2020-10-31'),  
('voz sin voz', '2019-07-22'),  
('ataques sin causa', '2018-03-13');
```

```
insert into investigacion ('id_objeto, resumen, fecha) values  
('1', 'analisis energia residual', '2021-01-05'),  
('2', 'manejo estaciones opticas', '2022-04-20'),  
('3', 'lecturas electromagneticas', '2023-06-15');
```

```
insert into objetos_eventos (id_objeto, id_evento) values  
('1,1'),  
('1,2'),  
('2,2'),  
('3,3'),  
('1,3');
```

```
insert into curador_objeto (id_curador, id_objeto) values  
('1,1'),  
('2,2'),  
('3,3'),  
('1,3');
```

```
-- Elimina la relacion objeto-evento entre 1 y 2  
delete from objeto_evento where id_objeto = 1 and id_evento = 2;  
-- Elimina la investigacion del objeto 2  
delete from investigacion where id_objeto = 2;  
-- borra el curador 3 y sus asignaciones  
delete from curador_objeto where id_curador = 3;  
-- actualiza objeto-curador donde el objeto 1  
update objeto set peligrosidad del objeto 1  
-- corrige fecha en investigacion 1  
update investigacion set fecha = '2021-02-01' where id_objeto = 1;  
-- renombra evento 2  
update evento set descripcion = 'eco sin aviso' where id_evento = 2;  
-- inner join: investigaciones con nombre de objetos  
select o.nombre, i.descripcion, i.fecha  
from investigacion i  
inner join objeto o on id_objeto = o.id_objeto;  
-- select o.nombre e.descripcion  
from objeto o  
left join evento e on o.id_objeto = e.id_evento  
group by o.nombre  
-- right join curador_objeto on curador_objeto = o.id_objeto;  
right join objeto o on curador_objeto = o.id_objeto;  
-- count: cuantas veces por objeto  
select o.nombre count, count(id_evento) as total_eventos  
from objeto o  
left join evento e on o.id_objeto = e.id_evento  
group by o.nombre;
```

```
- sum: total de invasiones por objetos
  select o.nombre, count(o.id) as total_inves
  from objeto o
  join invasacion i on o.id_objeto = i.id_objeto
  group by o.nombre;
- avg: promedio de eventos por objetos
  select avg(c!_cnt) as avg_eventos from (
    select count(c) as cnt
    from objeto o
    join evento_e o.evento
    group by id_objeto
  )t;
```

```
- max: objetos con mas eventos (selección)
  select nombre_cnt from (
    select o.nombre, count(o.evento) as cnt
    from objeto o
    join evento_e o.evento
    on o.id_objeto = o.evento.id_objeto
    group by o.nombre
  )sub
  where cnt = C
  select max(cnt) from C
  select count(*) as cnt from objeto_e
  group by id_objeto
);
```

```
- min: objetos con menos eventos
  select nombre_cnt from (
    select o.nombre, count(o.evento) as cnt
    from objeto o
    left join objeto_e oe on o.id_objeto = oe.id_objeto
    group by o.nombre
  )sub
  where cnt = C
  select min(count) from C
  select count(*) as cnt from objeto_e
  group by id_objeto
);
```

-- Crear base de datos
Create database empresa_envios;

-- Crear tablas

```
create table cliente (
    id_cliente int primary key auto_increment,
    nombre varchar(100),
    telefono varchar(20),
    direccion varchar(150),
);
```

```
create table Paquete (
    id_paquete int primary key auto_increment,
    descripcion varchar(150),
    peso decimal(5,2),
    dimensiones varchar(50)
);
```

```
create table repartidor (
    id_repartidor int primary key auto_increment,
    nombre varchar(100),
    zona varchar(50),
);
```

```
create table nota (
    id_nota int primary key auto_increment,
    origen varchar(100),
    destino varchar(100),
    distancia float decimal(6,2)
);
```

```
create table estado_envio (
    id_estado int primary key auto_increment,
    descripcion varchar(50)
);
```

```
create table envio (
    id_envio int primary key auto_increment,
    cliente int,
    paquete int,
    repartidor int,
    ruta int,
    estado int,
    fecha_entrega date time,
    foreign key (id_cliente) references cliente (id_cliente),
    foreign key (id_paquete) references paquete (id_paquete),
    foreign key (id_repartidor) references repartidor (id_repartidor),
    foreign key (id_estado) references estado (id_estado),
    foreign key (id_estado) references estado_envio (id_estado_envio)
);
```

Create table reportador_ruta (id_reportador int primary key auto_increment,
id_ruta int foreign key (id_reportador) references reportador(id_reportador),
foreign key (id_ruta) references ruta(id_ruta))

2) - insert

Insert into cliente (nombre, telefono, dirección) values
'Juan Perez', '3112345678', 'cra 10 # 65-67',
'Casa torres', '3100088776', 'calle 12 # 34-56',
'Carlos', '3121123456', 'av. 60 # 22-33';

Insert into paquete (descripcion, peso, dimensiones) values
'Cicada con libro', 3.2, '40x30x26',
'Soda con documento', 0.3, '30x20x10',
'Electrodoméstico', 12.5, '60x50x40');

Insert into reportador (nombre, zona) values
'Carola Gomez', 'norte',
'David Sanchez', 'sur',
'Camila Ospina', 'centro');

Insert into ruta (origen, destino, distancia_km) values
'Bodega norte', 'zona A', 12.5, '20.5',
'Bodega sur', 'zona B', 15.0, '25.0',
'Bodega centro', 'zona C', 8.0, '13.0');

Insert into estado_envio (descpcion) values
'en tramo',
'en regreso',
'crecasado');

Insert into envio (cliente_id, paquete_id, reportador_id, ruta_id) values
(1, 1, 1, 1),
(2, 2, 2, 2),
(3, 3, 3, 3);

Insert into envio (cliente_id, paquete_id, reportador_id, ruta_id) values
(1, 1, 1, 2),
(2, 2, 1, 2),
(3, 3, 1, 2);
(4, 4, 2, 1),
(5, 5, 3, 1);
(6, 6, 1, 3),
(7, 7, 2, 3),
(8, 8, 3, 2);

-- delete
delete from envio where id_envio = 3;
delete from paquete where id_paquete = 3;

-- update
update reportador
set zona = 'Zona Centro'
where id_reportador = 3;

-- update ruta
set distancia_km = 30.0
where id_ruta = 2;
update estadio_envio
set descripcion = 'en regreso a cliente'
where id_estadio = 2;

Verde ruta
8 et distancias Km = 30.0

update cliente
set dirección = 'cra 40 # 45-98',
where id_cliente = 1;

--inner join

select

e.id_envelope

c.nombre as cliente

p.descripcion as paquete

r.nombre as repartidor

r.nombre as repartidor

r.destino

es.descripcion as estado,

e.fecha_envio

e.fecha_entrega

from envio as e

inner join cliente as c on e.id_cliente = c.id_cliente

inner join paquete as p on e.id_paquete = p.id_paquete

inner join repartidor as r on e.id_repartidor = r.id_repartidor

inner join ruta as r1 on e.id_ruta = r1.id_ruta

inner join estado_envio as es on e.id_estado = es.id_estado

--left join

select

r1.nombre

r1.destino

r1.id_repartidor as repartidor

from ruta as r1 on r1.id_ruta = r.id_ruta

left join repartidor as rcp on r1.id_repartidor = rcp.id_repartidor

--right join

select

rep.nombre as repartidor,

r1.nombre

r1.destino

from repartidor as rep

right join repartidor as r1 on rep.id_repartidor =

r1.id_repartidor

left join ruta as r1 on r1.id_ruta = r.id_ruta;

--sum

sum(p.peso) as peso_total

from paquete as p;

--avg

select avg(cantidad_ruta.distancia_km) as promedio_distancia

from ruta as r;

--count distinct

select count(distinct descripcion) as estados_diferentes

from estados_envio;

--max

select descripcion, peso

from paquete where peso = (select max(peso) from paquete);

-- min
Select origin - destination km
From ref. table - km = (select min cost path - km) from order;

-- Count groups

select es.descripcion as es_fato, count(cel_id_envo) as cont_dados from envio as envio inner join estada_envo as est on e.id_estado = es.id_estado group by es.descripcion;

- having sole r. number as reporter, count circ. rate) as total ticket
from reporter as r. number as reporter - rate as in on r. lch reporter
group by r. number
having total nfc > 7.2

~~-> Subscriptions in
Select number
from Clientes
here id-cliente in
From envio~~

Subsidiary Company and all
selected products
from Client
where Client
selected selected Client
from euro as
inner gen es facturado en pesos, on e. col - estando - el de esta
when es. Los precios
like j, en la paga de

Create database agenciasviajesdb;
use agenciasviajesdb;

-- tablas

Create table cliente (
id_cliente int auto-increment primary key,
nombre varchar(100) not null,
email varchar(100)
telefono varchar(20))

Create table destino (
id_cliente int auto-increment primary key,
ciudad varchar(100) not null,
pais varchar(100) not null
)

Create Table guia (
id_guia int auto-increment primary key,
nombre varchar(100) not null,
precio decimal(10,2),
descripcion text,
id_destino int,
id_cliente int,
leguia int key (id_cliente)
references destino(id_cliente)
on delete cascade
on update cascade

foreign key (id_guia)
references guia (id_guia)
on delete set null
on update cascade

Create table paquete (
id_paquete int auto-increment primary key,
nombre varchar(100) not null,
precio decimal(10,2),
descripcion text,
id_destino int,
id_cliente int,
leguia int key (id_cliente)
references destino(id_cliente)
on delete cascade
on update cascade

create table paquete_actuacion (id_paquete int auto-increment primary key,

id_paquete_actuacion int auto-increment primary key,
id_paquete int not null
id_actuacion int not null
foreing key (id_paquete) references paquete(id_paquete)

on update cascade
foreign key (id_actuacion) references actuacion(id_actuacion)
on delete cascade
on update cascade

)

create table reserva (id_reserva int auto-increment primary key,
id_cliente int not null,
id_paquete int not null,
fecha_reserva date,
personal int,
foreign key (id_cliente) references cliente(id_cliente),
on delete cascade
on update cascade,

foreign key (id_paquete)
references paquete(id_paquete)
on delete cascade
on update cascade

)

insert into Cliente (Nombre, email, telefono) values
(Laura, 'laura@mail.com', '3112345678'),
(Carlos Vomero, 'carlos@mail.com', '3102826343'),
(Celia Perez, 'celia@mail.com', '3104567890');

insert into destino (ciudad, pais) values
(Paris, 'francia'),
(Tokio, 'japon'),
(Cartagena, 'colombia');

insert into grupo (nombre, id_oma) values
('ptis toros', 'espanol'),
('emily white', 'ingles'),
('christopher', 'ingles'),
('nakanaka', 'japones');

insert into paquete (nombre, precio) values
(destino, grupo) values
(('romance', 'en Paris'), 3500.00),
(('cultural', 'en Tokio'), 4200.00),
(('sociedad', 'en Cartagena'), 1200.00),
(('sol', 'en Madrid'), 2000.00);

insert into actuacion (numero, tpo) values
(('city tour', 'historico'), 1.00),
(('cena romantica', 'gourmet'), 1.00),
(('esnorkele 20sh', 'aventura'), 1.00),
(('museo del louvre', 'cultural'), 1.00);

insert into reserva (id_paquete, id_cliente, fecha_reserva, personal) values
(('sol', 'en Madrid'), 1, '2023-05-22', 1),
(('cena romantica', 'gourmet'), 2, '2023-06-07', 2),
(('esnorkele 20sh', 'aventura'), 3, '2023-07-13', 3),
(('museo del louvre', 'cultural'), 4, '2023-08-21', 4);

insert into paquete - cedula (C_id_paquete, id_cliente)

values ('2025-06-07', 122),

(1,1), (1,2), (1,5),

(2,1), (2,2), (2,7),

(3,3), (3,4);

select r.id_reserva, c.nombre as cliente,
p.nombre as paquete,
r.fecha_reserva,
r.personas
from reserva r
join cliente c on r.id_cliente = c.id_cliente,
join paquete p on r.id_paquete = p.id_paquete;

-- paquetes y reservas (enlazo sin reservar)

select p.nombre as paquete,
r.id_cliente as cliente
c.nombre as cliente
from paquete p
left join reserva r on p.id_paquete = r.id_paquete
left join cliente c on r.id_cliente = c.id_cliente;

select p.nombre as paquete,
count(r.id_reserva) as total_reservas

from paquete p
left join reserva r on p.id_paquete = r.id_paquete
group by p.nombre;

select sum(personas) as total_vigencias
from reserva;

select avg(precio) as precio_promedio
from paquete;

select nombre, precio
from paquete
where precio = (select max(precio) from paquete);

select g.nombre as ciudad
from grupo g
join paquete p on g.id_ciudad = p.id_ciudad
group by g.nombre
having total_paquetes > 1;

Result Client number C. number
join research r on C. id - Client = r. id - package
join package p on r. id - package = p. id - defining
where r. id - defining
update client set phone = 3119999999, where id-client;
update package set price * 110 where id-package = 2;
update package set id = 110 where id-package = 2;
update package set id = 110 where id-package = 2;
update package set id = 110 where id-package = 2;
update package set id = 110 where id-package = 2;
update package set id = 110 where id-package = 2;
update package set id = 110 where id-package = 2;

Create database ConjuntoResidencial;

use ConjuntoResidencial;

Create table apartamento (

- id_apartamento int primary key auto-increment,
- numero varchar(10),
- telefono varchar(20),
- correo varchar(50)

);
Create table residente (

- id_residente int primary key auto-increment,
- nombre varchar(100),
- telefono varchar(20),
- departamento int foreign key (id_departamento) references apartamento (id_apartamento),

);
Create table Pago (

- id_pago int primary key auto-increment,
- id_residente int,
- fecha date,
- concepto varchar(100),
- valor decimal(10,2)

);
foreign (id_residente) references residente (id_residente);
Create table visita (

- id_visita int primary key auto-increment,
- id_residente int,
- fecha_ingreso date,
- foreign key (id_apartamento) references apartamento (id_apartamento),
- foreign key (id_residente) references residente (id_residente)

);
Create table apartamento_residente (

- id_apartamento_residente int primary key auto-increment,
- id_residente int,
- foreign key (id_residente) references residente (id_residente),
- foreign key (id_apartamento) references apartamento (id_apartamento)

);
foreign key (id_residente) references residente (id_residente);
D.p.s0
lat

```
create table registro_visita (id_registro int primary key auto_increment,
id_visitante int,
id_departamento int,
fecha datetime,
motivo varchar(100),
foreign key (id_visitante) references visita (id_visitante),
foreign key (id_departamento) references departamento (id_departamento));
```

```
insert into apartamento (numero, piso, calle) values ('101', 'A', 1), ('202', 'B', 2), ('303', 'C', 3);
```

```
insert into residente (nombre, telefono, correo) values ('Laura Gomez', '3124567890', 'laura@mail.com'), ('Carles Ramon', '3109876543', 'carles@gmail.com'), ('Morana Torres', '3111223333', 'morana@gmail.com');
```

```
insert into parquero (numero, calle, id_apartamento) values ('P01', 'Calle 10', 1), ('P02', 'Major', 2), ('P03', 'Carolina', 3);
```

```
insert into pago (id_residente, Precio, Concepto, valor) values ('Carola Lopez', '1234567890', 'Administracion', 150000), ('Ana Martinez', '987654321', 'Administracion', 120000), ('Paula Ruiz', '4567890123', 'Administracion', 150000);
```

```
insert into visita (nombre, telefono) values ('Andrea Lopez', '1234567890'), ('Ana Martinez', '987654321'), ('Paula Ruiz', '4567890123');
```

```
insert into apartamento_residente (id_apartamento, id_residente) values ('1', '1'), ('2', '2'), ('3', '3');
```

```
delete from registro_visita where id_registro = 3;
delete from visita where id_visitante = 2;
delete from parquero where id_parquero = 2;
update apartamento
set nombre = 'Laura Gomez'
where id_departamento = 2;
```

```
update apartamento
set piso = 4
where id_departamento = 2;
```

update pago
set valor = 135000
where id_pago = 2;

update visita
set nombre = 'andrea lópez'
where id_visita = 7;

Select v.nombre as visitante, a.numero as apartamento,
n.fecha_rv as fecha_rv
from registro as v
inner join visita as rv on v.id_visita = rv.id_visita
inner join apartamento as a on rv.id_apartamento =
a.id_apartamento;

Select a.numero as apartamento, r.nombre as residente,
ar.fecha_ingreso
from apartamento as a
left join apartamento as residente as ar on a.id_apartamento =
residente.ag
left join residente as r on ar.id_residente = r.id_residente;

Select sum(valor) as total_recaudado from pago;

Select avg(distinct concepto) as conceptos_distintos from pago;

Select count(DISTINCT concepto) as conceptos_distintos from pago;

Select nombre, max(valor) as mayor_pago
from pago
group by nombre;

Select nombre as valor
from pago
inner join residente as r on p.id_residente = r.id_residente
where valor = (select min(valor) from pago);

Select a.numero as apartamento, count(ar.id_residente)
as total_visitas
from apartamento as a
left join visita as rv on a.id_apartamento =
rv.id_apartamento
group by a.numero;

Select nombre
from residente
where id_residente in (

Select id_residente
from pago
where valor > 140000

Select numero
from apartamento
where id_apartamento in (Select id_apartamento from
registro as visited acr on rv.id_visita = v.id_visita
where v.nombre = 'andrea lópez')