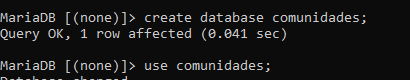
EJERCICIOS DE BASE DE DATOS

1.Directorio de comunidades

Se quiere desarrollar un directorio de comunidades para eso necesitamos guardar de cada comunidad el nombre, que tipo de comunidades (tecnológica, ciencias, social,etc), el país, estado y sus redes sociales, las comunidades en su mayoría realizan eventos o actividades de las cuales también se necesita tener registro, de cada evento que llevan a cabo será necesario guardar registro del nombre del evento o actividad, objetivo del evento, fecha y hora, medio por el cual será (zoom, presencial) y un detalle de ese medio, por último una lista de requerimientos de tal evento. Además de los eventos las comunidades necesitan saber con qué personas cuentan para ello se llevará registro de las personas, es necesario el nombre, dirección, a que se dedica, medio por el cuál conoció a la comunidad (facebook, por un amigo, en la escuela,etc), y por qué ha elegido unirse a esa comunidad.

Creación de la base de datos



Diseño de la base de datos

create table direccion\_base(

id\_dire int(20) primary key,

pais varchar(30),

estado varchar(30),

municipio varchar(30),

ciudad varchar(30);

);

create table ubicacion(

id\_ubicacion int(20) primary key,

numero int(30),

codigopostal int(30),

id\_dire int(20),

foreign key(id\_dire) references direccion\_base(id\_dire));

create table redes(

id\_redes int(20) primary key,

nombre varchar(30),

enlace varchar(30));

create table tipocomunidad(

id\_tipocomu int(20) primary key,

tipo\_comunidad varchar(30));

create table comunidad(

id\_comunidad int(20)primary key,

nombre varchar(30),

id\_tipocomu int(20),

id\_dire int(20),

foreign key(id\_tipocomu) references tipocomunidad(id\_tipocomu),

foreign key(id\_dire) references direccion\_base(id\_dire));

create table redcom(

id\_redcom int(20) primary key,

id\_redes int(20),

id\_comunidad int(20),

foreign key(id\_redes) references redes(id\_redes),

foreign key(id\_comunidad) references comunidad(id\_comunidad));

create table evento(

id\_evento int(20) primary key,

nombre varchar(30),

objetivo varchar(50),

fecha datetime,

id\_comunidad int(20),

foreign key(id\_comunidad) references comunidad(id\_comunidad));

create table requerimiento(

id\_requerimiento int(20) primary key,

nombre varchar(30),

id\_evento int(20),

foreign key(id\_evento) references evento(id\_evento));

create table medios(

id\_medio int(20) primary key,

medio varchar(30));

create table medeve(

id\_medeve int(20) primary key,

detalle varchar(50),

id\_evento int(20),

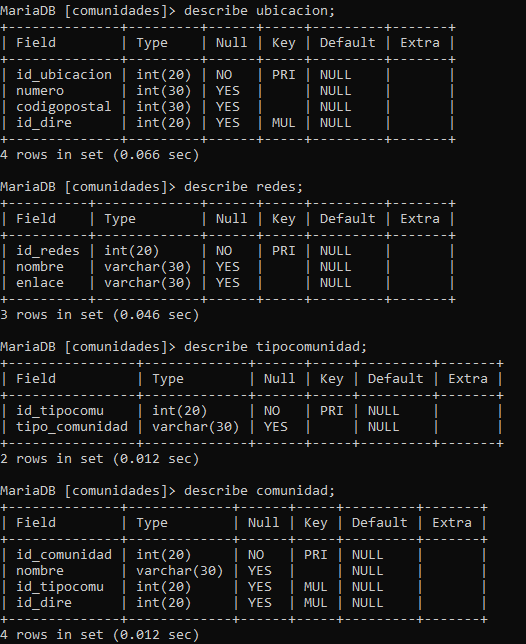
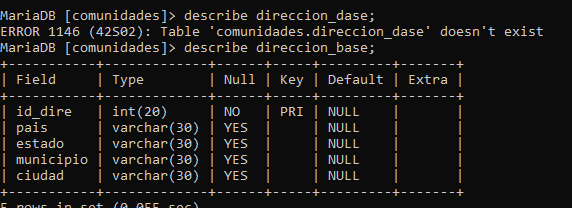
id\_medio int(20),

foreign key(id\_evento) references evento(id\_evento),

foreign key(id\_medio) references medios(id\_medio)

);

DICCIONARIO DE DATOS:



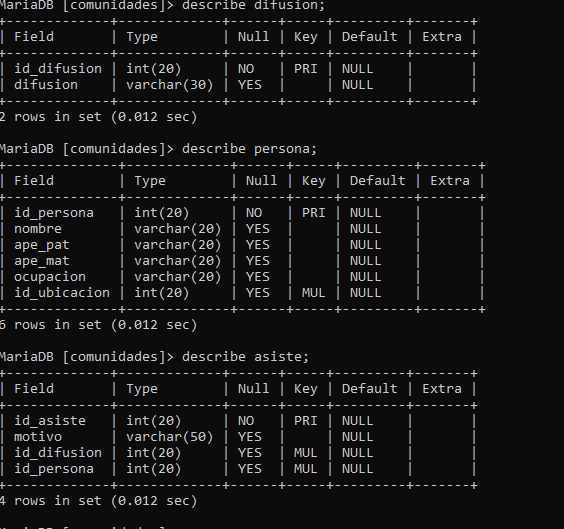
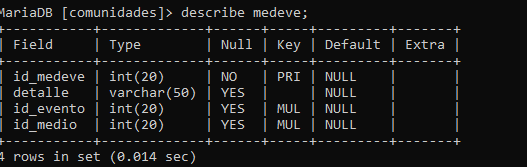
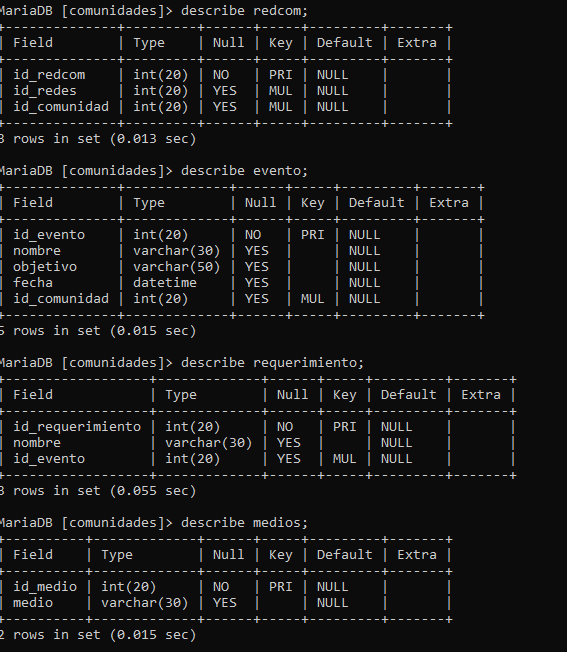
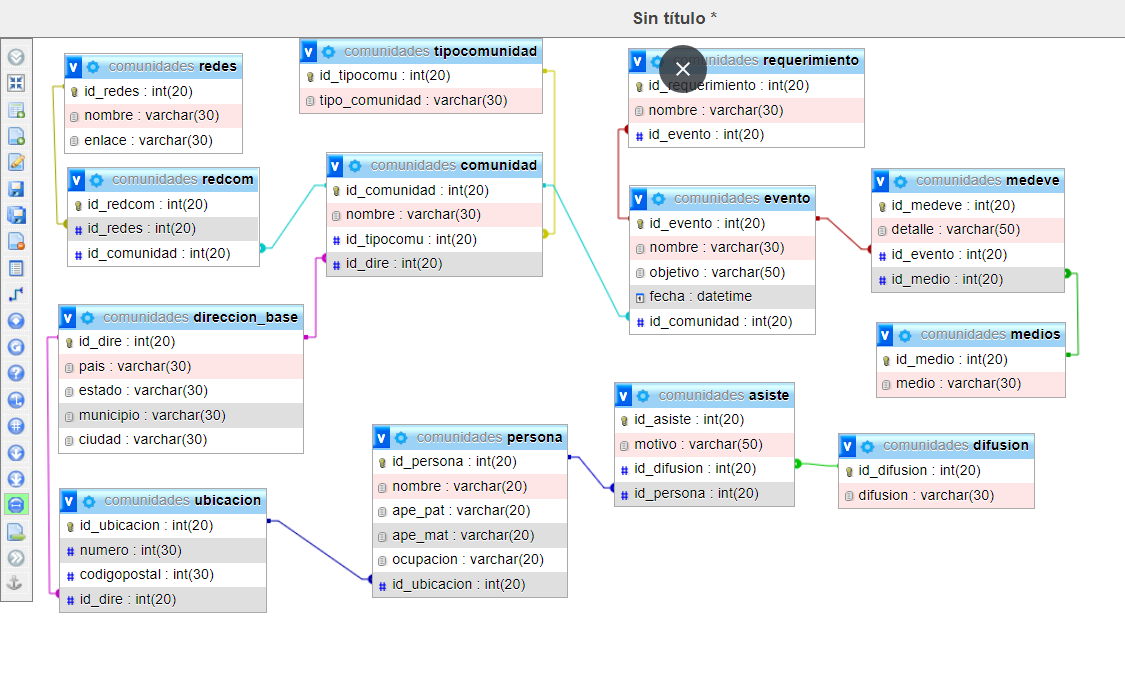


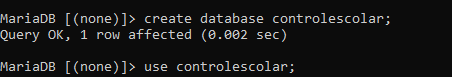
DIAGRAMA DE RELACIONES:



2.Control Escolar

Se requiere tener control de los alumnos, materias y profesores de una escuela. Cada que se realiza una inscripción de un alumno se deberá guardar el nombre del alumno, correo, dirección, curp, fecha de nacimiento, grado a cursar además se requiere tener registro de esa inscripción guardando la fecha en que se inscribió, medio por el cual fue inscrito (internet, presencial) en caso que haya sido presencial que persona atendió la inscripción esta persona debe ser un administrativo del plantel por lo cual además de tener un nombre debe de tener un cargo o puesto en la institución, una vez que se inscribe un alumno es necesario hacer la carga de materias estas materias pueden impartirlas distintos profesores, de cada materia se debe guardar el nombre, total de horas requeridas,calificación min. aprobatoria. De los profesores se debe tener nombre, dirección y correo.

Creación de la base de datos



Diseño de la base de datos

create table direccion\_base(

id\_dire int(20) primary key,

pais varchar(30),

estado varchar(30),

municipio varchar(30),

ciudad varchar(30));

create table ubicacion(

id\_ubicacion int(20) primary key,

numero int(30),

codigopostal int(30),

id\_dire int(20),

foreign key(id\_dire) references direccion\_base(id\_dire));

create table grado(

id\_grado int(20) primary key,

grado varchar(25));

create table persona(

id\_persona int(20) primary key,

nombre varchar(30),

apell\_pat varchar(30),

apell\_mat varchar(30));

create table alumno(

id\_alumno int(11) primary key,

correo varchar(25),

curp varchar(25),

fecha\_nat varchar(25),

id\_grado int(20),

id\_persona int(20),

id\_ubicacion int(20),

foreign key(id\_grado) references grado(id\_grado),

foreign key(id\_persona) references persona(id\_persona),

foreign key(id\_ubicacion) references ubicacion(id\_ubicacion));

create table medio\_ins(

id\_medio int(20) primary key,

medio varchar(30));

create table puesto(

id\_puesto int(20) primary key,

puesto varchar(30));

create table administrativo(

id\_administrativo int(20) primary key,

id\_persona int(20),

id\_puesto int(20),

foreign key(id\_persona) references persona(id\_persona),

foreign key(id\_puesto) references puesto(id\_puesto));

create table inscripcion(

id\_insc int(20) primary key,

fecha datetime,

id\_medio int(20),

id\_administrativo int(20),

id\_alumno int(20),

foreign key(id\_medio) references medio\_ins(id\_medio),

foreign key(id\_administrativo) references administrativo(id\_administrativo),

foreign key(id\_alumno) references alumno(id\_alumno));

create table materia(

id\_materia int(20) primary key,

nombre varchar(30),

horas int(30),

calificacion\_min int(50));

create table profesor(

id\_profesor int(20) primary key,

correo varchar(30),

id\_persona int(20),

id\_ubicacion int(20),

id\_puesto int(20),

foreign key(id\_persona) references persona(id\_persona),

foreign key(id\_ubicacion) references ubicacion(id\_ubicacion),

foreign key(id\_puesto) references puesto(id\_puesto));

create table mat\_pro(

id\_matpro int(20) primary key,

id\_materia int(20),

id\_profesor int(20),

foreign key(id\_materia) references materia(id\_materia),

foreign key(id\_profesor) references profesor(id\_profesor));

create table altamat(

id\_altamat int(20) primary key,

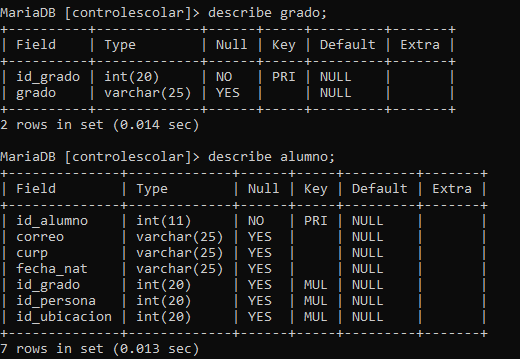
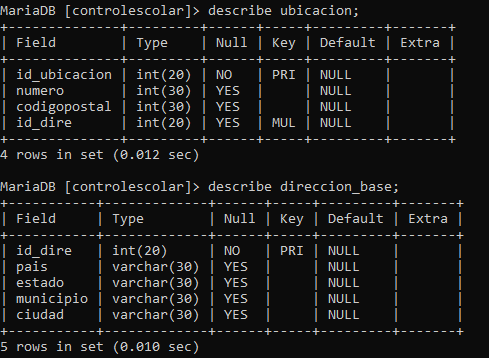
id\_matpro int(20),

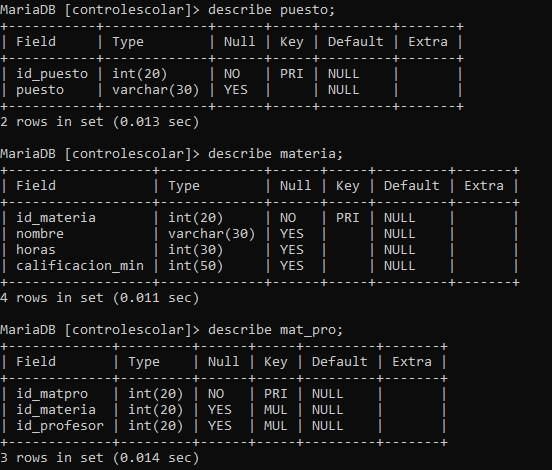
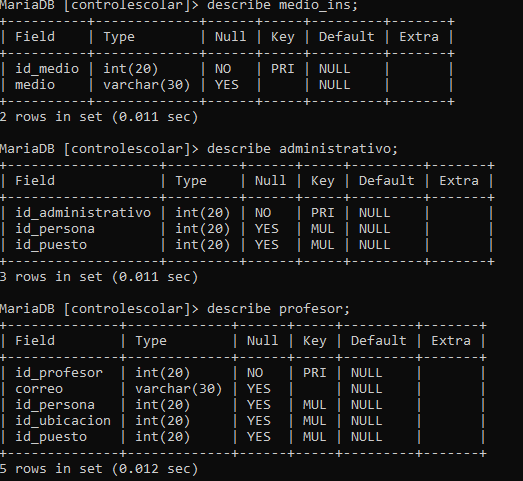
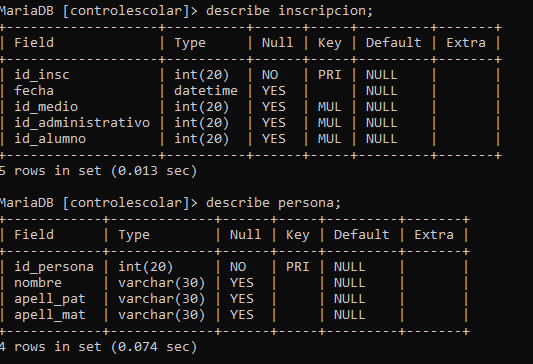
id\_alumno int(20),

foreign key(id\_matpro) references mat\_pro(id\_matpro),

foreign key(id\_alumno) references alumno(id\_alumno));

DICCIONARIO DE DATOS:





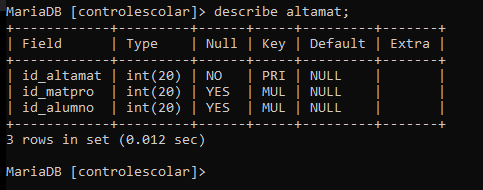


DIAGRAMA DE RELACIONES:

