contraseña root account:123456

usuario bd: controlasist

contraseña: ctrasis486

puerto: 3306

CREATE DATABASE `controlasistencia` /\*!40100 DEFAULT CHARACTER SET utf8 \*/;

CREATE TABLE `controlasistencia`.`evento` (

`idevento` INT NOT NULL AUTO\_INCREMENT,

`codigo` VARCHAR(6) NOT NULL,

`categoria` VARCHAR(20) NOT NULL,

`nombre` VARCHAR(250) NOT NULL,

`fecha\_inicio` DATETIME NOT NULL,

`fecha\_fin` DATETIME NOT NULL,

`espacios` INT NOT NULL,

`url\_fanpage` VARCHAR(500) NULL,

`nombre\_contacto` VARCHAR(300) NULL,

`correo\_contacto` VARCHAR(150) NULL,

`ubicacion` VARCHAR(3000) NULL,

PRIMARY KEY (`idevento`),

UNIQUE INDEX `codigo\_UNIQUE` (`codigo` ASC))

COMMENT = 'registro de los eventos';

CREATE TABLE `controlasistencia`.`persona` (

`idpersona` INT NOT NULL AUTO\_INCREMENT,

`identificacion` VARCHAR(25) NOT NULL,

`nombre` VARCHAR(250) NOT NULL,

`telefono` VARCHAR(45) NULL,

`correo` VARCHAR(100) NULL,

PRIMARY KEY (`idpersona`),

UNIQUE INDEX `identificacion\_UNIQUE` (`identificacion` ASC));

CREATE TABLE `controlasistencia`.`evento\_persona` (

`idevento\_persona` INT NOT NULL AUTO\_INCREMENT,

`idevento` INT NOT NULL,

`idpersona` INT NOT NULL,

PRIMARY KEY (`idevento\_persona`),

UNIQUE INDEX `idevento\_persona\_UNIQUE` (`idevento\_persona` ASC),

INDEX `FK\_EVENTOPERSONA\_PERSONA\_idx` (`idpersona` ASC),

INDEX `FK\_EVENTOPERSONA\_EVENTO\_idx` (`idevento` ASC),

CONSTRAINT `FK\_EVENTOPERSONA\_PERSONA`

FOREIGN KEY (`idpersona`)

REFERENCES `controlasistencia`.`persona` (`idpersona`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `FK\_EVENTOPERSONA\_EVENTO`

FOREIGN KEY (`idevento`)

REFERENCES `controlasistencia`.`evento` (`idevento`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

ALTER TABLE `controlasistencia`.`evento\_persona`

ADD UNIQUE INDEX `UNQ\_EVENTOPERSONA` (`idevento` ASC, `idpersona` ASC),

DROP INDEX `idevento\_persona\_UNIQUE` ;

CREATE DEFINER=`controlasist`@`%` PROCEDURE `updateEvento`(in pIdEvento int , in pCodigo varchar(6), in pCategoria varchar(20),

in pNombre varchar(250),in pEspacios int,in pUrlFanPage varchar(500),in pNombreContacto varchar(300), in pCorreoContacto varchar(150),

in pUbicacion varchar(3000),in pFechaInicio datetime,in pFechaFin datetime)

BEGIN

update evento e set e.codigo = pCodigo, e.categoria = pCategoria, e.nombre = pNombre, e.espacios = pEspacios,

e.url\_fanpage = pUrlFanPage, e.nombre\_contacto = pNombreContacto, e.correo\_contacto = pCorreoContacto, e.ubicacion = pUbicacion,

e.fecha\_inicio = pFechaInicio, e.fecha\_fin = pFechaFin

where idevento = pIdEvento;

END

CREATE DEFINER=`controlasist`@`%` PROCEDURE `insertEvento`(in pCodigo varchar(6),in pCategoria varchar(20),

in pNombre varchar(250), in pFechaInicio datetime, in pFechaFin datetime,

in pEspacios int, in pUrlFanPage varchar(500), in pNombreContacto varchar(300),

in pCorreoContacto varchar(150), in pUbicacion varchar(3000), out pIdEvento int)

BEGIN

insert into evento(codigo,categoria,nombre,fecha\_inicio,fecha\_fin,espacios,url\_fanpage,nombre\_contacto,correo\_contacto,ubicacion)

values (pCodigo,pCategoria,pNombre,pFechaInicio,pFechaFin,pEspacios,pUrlFanPage,pNombreContacto,pCorreoContacto,pUbicacion);

set pIdEvento = last\_insert\_id();

END

USE `controlasistencia`;

DROP procedure IF EXISTS `insertEventoPersona`;

DELIMITER $$

USE `controlasistencia`$$

CREATE DEFINER=`controlasist`@`%` PROCEDURE `insertEventoPersona`(in pIdEvento int, in pIdPersona int)

BEGIN

declare lCantPersonasMatriculadas int;

declare lCupoEvento int;

select count('X') into lCantPersonasMatriculadas

from evento\_persona ep

where ep.idevento = pIdEvento;

select e.espacios into lCupoEvento from evento e

where e.idevento = pIdEvento;

if (lCantPersonasMatriculadas < lCupoEvento) then

insert into evento\_persona(idevento,idpersona) values(pIdEvento,pIdPersona);

else

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'El evento no cuenta con espacios disponibles';

end if;

END$$

DELIMITER ;

CREATE TABLE `controlasistencia`.`asistencia` (

`idasistencia` INT NOT NULL AUTO\_INCREMENT,

`idevento\_persona` INT NOT NULL,

`fecha\_asistencia` DATETIME NOT NULL,

PRIMARY KEY (`idasistencia`),

INDEX `FK\_ASISTENCIA\_EVENTOPERSONA\_idx` (`idevento\_persona` ASC),

CONSTRAINT `FK\_ASISTENCIA\_EVENTOPERSONA`

FOREIGN KEY (`idevento\_persona`)

REFERENCES `controlasistencia`.`evento\_persona` (`idevento\_persona`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

USE `controlasistencia`;

CREATE OR REPLACE VIEW `view\_asistencia` AS

SELECT a.idasistencia,a.fecha\_asistencia,p.idpersona,ep.idevento ,p.identificacion,p.nombre

FROM asistencia a

inner join evento\_persona ep on ep.idevento\_persona = a.idevento\_persona

inner join persona p on p.idpersona = ep.idpersona

order by a.fecha\_asistencia desc;;

USE `controlasistencia`;

DROP procedure IF EXISTS `insertAsistencia`;

DELIMITER $$

USE `controlasistencia`$$

CREATE PROCEDURE `insertAsistencia` (in pIdEvento int,in pIdPersona int)

BEGIN

declare lExiste int;

declare lIdEventoPersona int;

SELECT

COUNT('X')

INTO lExiste FROM

evento\_persona ep

WHERE

ep.idevento = pIdEvento

AND ep.idPersona = pIdPersona;

if lExiste = 1 then

select ep.idevento\_persona into lIdEventoPersona

from evento\_persona ep

where ep.idevento = pIdEvento and ep.idPersona = pIdPersona;

insert into asistencia(idevento\_persona,fecha\_asistencia) values (lIdEventoPersona,sysdate());

else

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'La persona no se encuentra matriculada en el evento';

end if;

END$$

DELIMITER ;

CREATE DEFINER=`controlasist`@`%` PROCEDURE `updatePersona`(in pIdPersona int, in pIdentificacion varchar(25),

in pNombre varchar(250), in pTelefono varchar(45), in pCorreo varchar(100))

BEGIN

declare lExisteIdentificacion int;

select count('x') into lExisteIdentificacion

from persona p

where p.idPersona != pIdPersona and p.identificacion = pIdentificacion;

if (lExisteIdentificacion = 0) then

update persona p set p.identificacion = pIdentificacion,

p.nombre = pNombre,

p.telefono = pTelefono,

p.correo = pCorreo

where p.idpersona = pIdPersona;

else

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Ya existe una persona con la misma identificación.';

end if;

END

CREATE DEFINER=`controlasist`@`%` PROCEDURE `ìnsertPersona`(in pIdentificacion varchar(25),in pNombre varchar(250),

in pTelefono varchar(45), in pCorreo varchar(100), out pIdPersona int)

BEGIN

declare lExisteIdentificacion int;

select count('x') into lExisteIdentificacion

from persona p

where p.identificacion = pIdentificacion;

if(lExisteIdentificacion = 0) then

insert into persona(identificacion,nombre,telefono,correo)

values (pIdentificacion,pNombre,pTelefono,pCorreo);

set pIdPersona = last\_insert\_id();

else

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Ya existe una persona con la misma identificación.';

end if;

END