

UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL RESISTENCIA

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

**BASES DE DATOS**

Guía de Trabajos Prácticos: Unidad 3

Carpeta de Entrega

**Grupo 4**

Integrantes:

* **Andres**, Aldo Omar.
* **Bravo Pérez**, Agustín Nicolás.
* **Brites,** Agustín.
* **Sáez Franci**, Juliana Carla Desiree.
* **Salomón**, Hilel Mauricio.
* **Machuain**, Ezequiel.

*28 de Abril de 2023*

### 1) Defina los siguientes términos:esquema de una relación, esquema de una base de datos relacional, instancia de una relación, cardinalidad de una relación y grado de una relación.

Esquema de una relación:especificación que define un nombre de la relación, un nombre y tipo de dato para sus atributos, y sus restricciones.

Esquema de una base de datos relacional:conjunto de todas las especificaciones de las relaciones que forman parte de la base de datos.

Instancia de una relación:es una con columnas y un conjunto de filas, en un momento dado.

Cardinalidad de una relación:es el número de tuplas de la instancia de la relación.

Grado de una relación:es el número de campos de la relación.

***2)*** *¿Cuántas tuplas distINTas hay en una instancia de una relación de cardinalidad 22?*

Si la instancia tiene 22 tuplas, y la cardinalidad es la cantidad de tuplas de la instancia, entonces su cardinalidad es 22.

***3)*** *¿Provee el modelo relacional independencias física y lógica de datos?*

Dentro de SQL, se define el modelo conceptual al crear las tablas (CREATE TABLE table) y se modifica al alterar esas tablas. Este modelo determina restricciones de clave y de dominio que una vista no necesita saber (SELECT \* FROM table), y que pueden cambiar sin que la vista necesite cambiar.

Aún así, hay casos en los que la independencia lógica no es total.

Ejemplo:

CREATE VIEW nombres AS (SELECT nombre FROM usuarios);

ALTER TABLE usuarios DROP COLUMN nombre

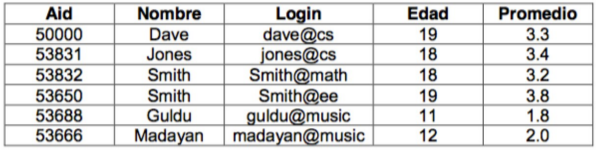
En este ejemplo, existía una vista que leía el atributo nombre de usuarios, que al eliminarse dejó de existir.

***4)*** *¿Cuál es la diferencia entre una clave candidata y la clave primaria de una relación? ¿Qué es una superclave?*

Una clave candidata de una relación es capaz de identificar unívocamente a cada tupla de la relación. La clave primaria es una clave candidata elegida para ser utilizada con ese propósito, que no puede ser nula. Además, es referenciada por las claves foráneas que referencian a la relación, y las bases de datos suelen ordenar físicamente las tuplas según su clave primaria.

Una superclave es una clave primaria formada por un conjunto de atributos. Para que una superclave sea distINTa a otra, es suficiente con que uno de los atributos que la conforman sea distINTo. Esto es útiles para entidades que no tienen claves candidatas, pero que la unión de dos o tres de sus atributos es siempre única.

***5)*** *Considere la instancia de la Alumnos mostrada en la siguiente figura:*

**

1. *Dé un ejemplo de un atributo (o conjunto de atributos) que no sea clave candidata, asumiendo que esta instancia es válida.Hay algún ejemplo de un atributo (o conjunto de atributos) que se pueda decir que es una clave candidata, asumiendo que ésta es una instancia válida?*

Atributos que NO son clave candidata:Nombre, Edad, Promedio.

Atributos que SÍ son clave candidata:Aid, Login.

***6)*** *¿Qué es una restricción de clave foránea? ¿Por qué son importantes tales restricciones? ¿Qué es INTegridad referencial?*

Una restricción de clave foránea indica que cada valor del campo restringido es idéntico a algún valor de la clave primaria de otra relación, a la cual está referenciando. Esto es importante porque nos permite representar vínculos entre relaciones, al asociar una con otra.

La INTegridad referencial busca asegurar que todas las claves foráneas referencien una clave primaria que efectivamente exista. En el caso hipotético de eliminar una tupla cuya clave primaria está siendo referenciada por una clave foránea de una tupla de otra tabla, esa segunda tupla quedaría referenciando a algo que no existe, perdiendo la INTegridad de los datos. Hay 4 formas posibles de mantener la INTegridad referencial:

- CASCADE:Si se elimina la tupla referenciada, también se elimina toda tupla que la está referenciando.

- NO ACTION:Se prohíbe la operación, de manera que no se puede eliminar la tupla referenciada sin antes eliminar todas las que la referencian.

- SET NULL:Se pone en nulo a todas las claves foráneas que referenciaban la tupla eliminada.

- DEFAULT:Se pone un valor por defecto a todas las claves foráneas que referenciaban la tupla eliminada.

***7)*** *Considere las siguientes relaciones:*

*Alumnos(ide:string, nombre:string, usuario:string, edad:INTeger, nota:real)*

*Profesores(idp:string, nombrep:string, sueldo:real)*

*Asignaturas(ida:string, nombrea:string, creditos:INTeger)*

*Aulas(nau:INTeger, direccion:string, capacidad:INTeger)*

*Matriculados(ide:string, ida:string, curso:string)*

*Imparte(idp:string, ida:string)*

*Impartida en(ida:string, nau:INTeger, hora:string)*

*1. Liste todas las claves foráneas de estas relaciones.*

*2. Dé un ejemplo válido de una restricción que involucre a una o más de estas relaciones y que no sea una restricción de clave primaria o clave foránea.*

Claves foráneas:Matriculados(ide, ida), Imparte(idp, ida), Impartida en(ida, nau).

Otras restricciones que involucran a estas relaciones son restricciones de dominio:los tipos de datos de sus campos.

***8)*** *Conteste sINTéticamente los siguientes ítems. Los mismos están basados en el siguiente esquema relacional:*

*Emp(eid:INTeger, ename:string, age:INTeger, salary:real)*

*Works(eid:INTeger, did:INTeger, pct time:INTeger)*

*Dept(did:INTeger, dname:string, budget:real, managerid:INTeger)*

1. *Dé un ejemplo de clave foránea que involucre a la relación Dept. ¿Cuáles son las opciones para forzar esta restricción cuando un usuario INTenta borrar una tupla en Dept?*
2. *Escriba las sentencias SQL apropiadas para crear las relaciones anteriores, incluyendo todas las restricciones de INTegridad de clave primaria y foránea.*

La clave foránea Works(did) referencia la clave principal Dept(did). Cuando una tupla de Dept es borrada, hay 3 alternativas:

- En cascada:se borran también las tuplas de Work que la referencian.

- Default:a las claves foráneas se les asigna un valor por defecto.

- Sin acción:la operación falla. Es la alternativa por defecto.

La cuarta alternativa de asignar un valor nulo no nos sirve porque la clave foránea Works(did) también es una clave primaria de Works, y una clave primaria no puede ser nula.

MySQL:

CREATE TABLE Emp(

eid INT,

ename VARCHAR(255) NOT NULL,

age INT,

salary DECIMAL(10,2),

PRIMARY KEY (eid)

);

CREATE TABLE Dept(

did INT PRIMARY KEY,

dname VARCHAR(255) NOT NULL,

budget DECIMAL,

managerid INT,

FOREIGN KEY(managerid) REFERENCES Emp(eid)ON DELETE SET NULL

);

CREATE TABLE Works(

eid INT,

did INT,

pct time INT,

PRIMARY KEY(eid, did),

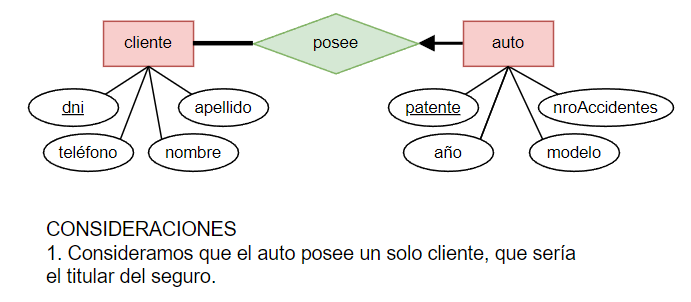
FOREIGN KEY (eid) REFERENCES Emp(eid) ON DELETE CASCADE,

FOREIGN KEY (did) REFERENCES Dept(did) ON DELETE CASCADE

);

***9)*** *Teniendo en cuenta los escenarios planteados en la unidad II (modelo entidad – relación), transformar cada DER obtenido en un esquema relacional en el que se han de indicar las claves primarias, las claves foráneas y valores no nulos.*

***10)*** *En base a los modelos relacionales surgidos del ejercicio anterior, escriba las sentencias SQL apropiadas para crearlas, incluyendo todas las restricciones de INTegridad de claves primaria y foráneas, así como cualquier otra restricción de verificación (checks), aserciones o disparadores (triggers) necesarios para captar toda la semántica posible del enunciado.*

1. 

9)

**cliente**(dni:INT, tel:INT, nombre:str, apellido:str)

CP:dni

CNN:nombre, apellido

**auto**(clienteDni:INT, patente:str, modelo:str, nroAccidentes:INT)

CP:patente

CF:clienteDni→Cliente(dni)

CNN:clienteDni

10)

CREATE TABLE **cliente**(

dni INT PRIMARY KEY,

tel INT,

nombre VARCHAR(50) NOT NULL,

apellido VARCHAR(50) NOT NULL,

);

CREATE TABLE **auto**(

clienteDni INT,

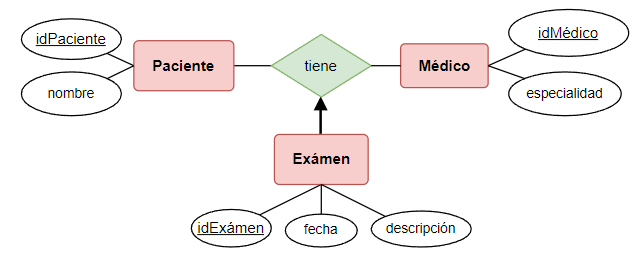
nroAcidentes INT,

modelo VARCHAR(50),

patente VARCHAR(50) PRIMARY KEY,

FOREIGN KEY (clienteDni) REFERENCES Cliente(dni)

);

2. 

9)

**paciente**(idPaciente:INT, nombre:str)

CP:idPaciente

CNN:nombre

**medico**(idMédico:INT, especialidad:str)

CP:idMédico

CNN:especialidad

**examen**(idExámen:INT, descripción:str, fecha:date, idPaciente:INT, idMédico:INT)

CP:idExámen

CF:idPaciente→Paciente(idPaciente), idMédico→Médico(idMédico)

CNN:idMédico, idPaciente, fecha

10)

CREATE TABLE **paciente**(

idPaciente INT PRIMARY KEY,

nombre VARCHAR(50) NOT NULL

);

CREATE TABLE **medico**(

idMedico INT PRIMARY KEY,

especialidad VARCHAR(40) NOT NULL

);

CREATE TABLE **examen**(

idExamen INT PRIMARY KEY,

fecha DATE NOT NULL,

descripcion TEXT,

idPaciente INT NOT NULL,

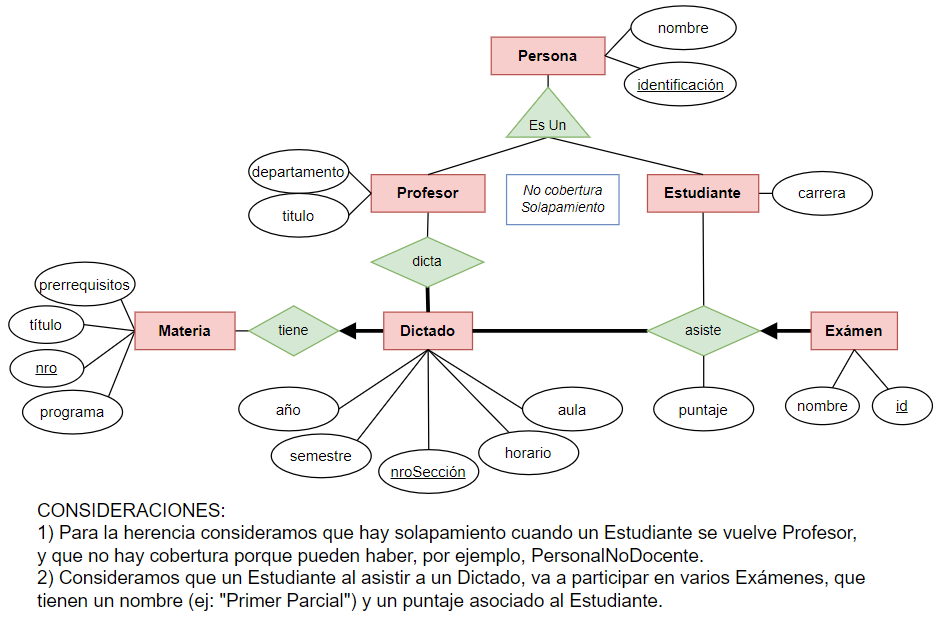
idMedico INT NOT NULL,

FOREIGN KEY (paciente) REFERENCES paciente(idPaciente),

FOREIGN KEY REFERENCES medico(idMedico)

)

3.



9)

**Persona**(id:INT, nombre:VARCHAR)

CP:id

CNN:nombre

**Profesor**(id:INT, titulo:VARCHAR, departamento:VARCHAR)

CP:id

CF:id→Persona(id)

CNN:titulo, departamento

**Estudiante**(id:INT, carrera:VARCHAR)

CP:id

CF:id→Persona(id)

CNN:carrera

**Materia**(nro:INT, titulo:VARCHAR, prerequisitos:VARCHAR, programa:VARCHAR)

CP:nro

CNN:titulo

**Dictado**(nroSeccion:INT, nroMateria:INT, año:INT, semestre:INT, horario:VARCHAR, aula:VARCHAR)

CP:nroSeccion

CF:nroMateria→Materia(nro)

CNN:nroMateria

**Dicta**(nroSeccion:INT, idProfesor:INT)

CP:nroSeccion, idProfesor

CF:idProfesor→Profesor(id), nroSeccion→Dictado(nroSeccion)

**Examen**(id:INT, idEstudiante:INT, nroSeccion:INT, nombre:VARCHAR, puntaje:INT)

CP:id

CF:idEstudiante→Estudiante(id), nroSeccion→Dictado (nroSeccion)

CNN:idEstudiante, nroSeccion, nombre

10)

CREATE TABLE **persona**(

id INT PRIMARY KEY,

nombre VARCHAR(40) NOT NULL

);

CREATE TABLE **profesor**(

id INT PRIMARY KEY,

titulo VARCHAR(40) NOT NULL,

departamento VARCHAR(40) NOT NULL,

FOREIGN KEY (id) REFERENCES persona(id)

);

CREATE TABLE **estudiante**(

id INT PRIMARY KEY,

carrera VARCHAR(3) NOT NULL,

FOREIGN KEY (id) REFERENCES persona(id)

);

CREATE TABLE **materia**(

nro INT PRIMARY KEY,

titulo VARCHAR (40) NOT NULL,

prerrequisito TEXT,

programa TEXT

);

CREATE TABLE **dictado**(

nroSeccion INT PRIMARY KEY,

nroMateria INT NOT NULL

año INT,

semestre INT,

horario VARCHAR(5),

aula VARCHAR(10),

FOREIGN KEY (nroMateria) REFERENCES materia(nro)

);

CREATE TABLE **dicta**(

nroSeccion INT,

idProfesor INT,

PRIMARY KEY(nroSeccion, idProfesor),

FOREIGN KEY (nroSecccion) REFERENCES materia(nro),

FOREIGN KEY (idProfesor) REFERENCES profesor(id)

);

CREATE TABLE **examen**(

id INT PRIMARY KEY,

idEstudiante INT NOT NULL,

nroSeccion INT NOT NULL,

nombre VARCHAR(40) NOT NULL,

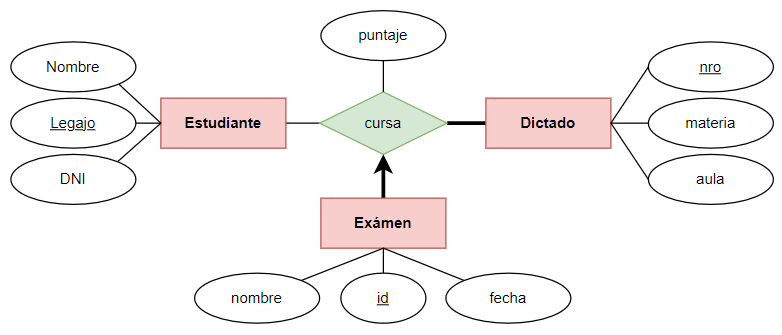
puntaje INT,

FOREIGN KEY (idEstudiante) REFERENCES estudiante(id),

FOREIGN KEY (nroSeccion) REFERENCES dictado(nroSeccion)

);

4. a)



9)

**Estudiante**(legajo:INT, dni:INT, nombre:VARCHAR)

CP:legajo

CNN:dni, nombre

**Dictado**(nro:INT, aula:VARCHAR, materia:VARCHAR)

CP:nro

CNN:materia, aula

**Examen**(id:INT, legajoEstudiante:INT, nroDictado:INT, fecha:date, nombre:VARCHAR, puntaje:INT)

CP:id

CF:legajoEstudiante→Estudiante(legajo), nroDictado→Dictado(nro)

CNN:legajoEstudiante, nroDictado, nombre, fecha

10)

CREATE TABLE **estudiantes**(

legajo INT PRIMARY KEY,

nombre VARCHAR(50) NOT NULL,

dni INT NOT NULL

);

CREATE TABLE **dictado**(

nro INT PRIMARY KEY,

materia VARCHAR(40) NOT NULL,

aula VARCHAR(10) NOT NULL

);

CREATE TABLE **examen**(

id INT PRIMARY KEY,

fecha DATE NOT NULL,

nombre VARCHAR(40) NOT NULL,

puntaje INT,

idDictado INT NOT NULL,

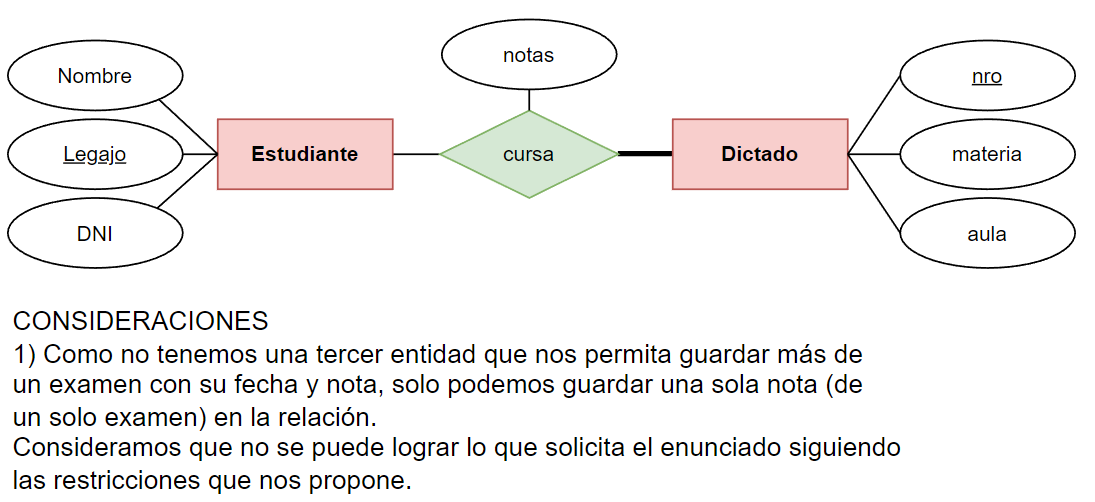
legajoEstudiante INT NOT NULL,

FOREIGN KEY (idDictado) REFERENCES dictado(nro),

FOREIGN KEY (legajoEstudiante) REFERENCES estudiante(legajo)

)

4. b)



9)

**Estudiante**(legajo:INT, dni:INT, nombre:VARCHAR)

CP:legajo

CNN:dni, nombre

**Dictado**(nro:INT, aula:VARCHAR, materia:VARCHAR)

CP:nro

CNN:aula, materia

**Cursa**(legajoEstudiante:INT, nroDictado:INT, notas:VARCHAR)

CP:legajoEstudiante, nroDictado

CF:nroDictado→Dictado(nro), legajoEstudiante→Estudiante(legajo)

10)

CREATE TABLE **estudiante**(

legajo INT PRIMARY KEY,

nombre VARCHAR(50) NOT NULL,

dni INT NOT NULL

);

CREATE TABLE **dictado**(

nro INT PRIMARY KEY,

materia VARCHAR(40) NOT NULL,

aula VARCHAR(10) NOT NULL,

);

CREATE TABLE **cursada**(

legajoEstudiante INT,

nroDictado INT,

notas VARCHAR(50),

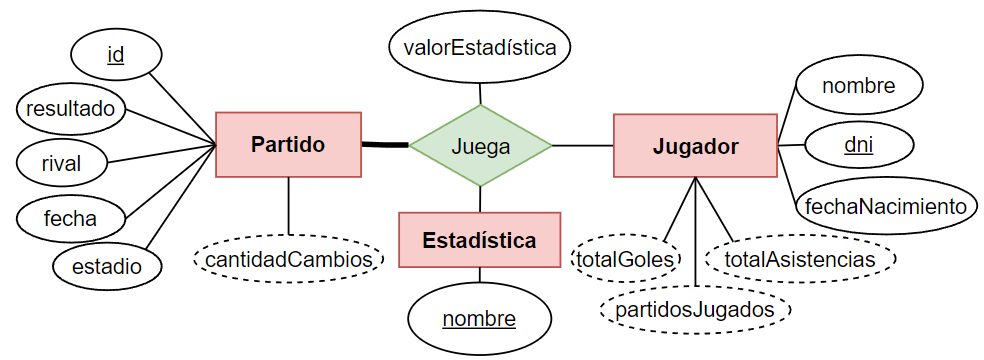
PRIMARY KEY (legajoEstudiante, nroDictado),

FOREIGN KEY (nroDictado) REFERENCES dictado(nro),

FOREIGN KEY (legajoEstudiante) REFERENCES estudiante(legajo)

);

5.



9)

**Partido**(id:INT, resultado:INT, rival:VARCHAR, fecha:date, estadio:VARCHAR)

CP:id

CNN:rival, fecha, estadio

**Estadistica**(nombre:VARCHAR)

CP:nombre

**Jugador**(dni:INT, fechaNacimiento:date, nombre:VARCHAR)

CP:dni

CNN:nombre, fechaNacimiento

**Juega**(nombreEstadistica:VARCHAR, dniJugador:INT, valorEstadística:INT, idPartido:INT)

CP:idPartido, idJugador, nombreEstadistica

CF: nombreEstadistica→Estadistica(nombre), dniJugador→Jugador(dni), idPartido→Partido(id)

CNN:valorEstadística

10)

CREATE TABLE **partido**(

id INT PRIMARY KEY,

resultado INT,

rival VARCHAR(40) NOT NULL,

fecha DATE NOT NULL,

estadio VARCHAR(40) NOT NULL

);

CREATE TABLE **estadistica**(

nombre VARCHAR (40) PRIMARY KEY

);

CREATE TABLE **jugador**(

dni INT PRIMARY KEY,

fechaNacimiento DATE NOT NULL,

nombre VARCHAR(50) NOT NULL

);

CREATE TABLE **Juega**(

idPartido INT,

nombreEstadistica VARCHAR(40),

dniJugador INT,

valorEstadistica INT NOT NULL,

PRIMARY KEY(idPartido, nombreEstadistica, dniJugador),

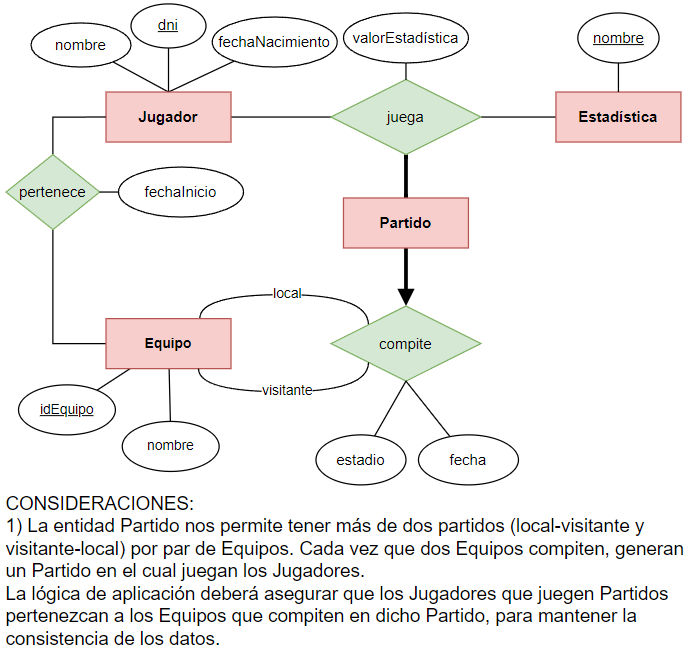
FOREIGN KEY (idPartido) REFERENCES partido(id),

FOREIGN KEY (nombreEstadistica) REFERENCES estadistica(nombre),

FOREIGN KEY (dniJugador) REFERENCES jugador(dni)

)

6.



9)

**estadistica**(nombre:VARCHAR)

CP:nombre

**equipo**(id:INT, nombre:VARCHAR)

CP:id

CNN:nombre

**jugador**(dni:INT, fechaNacimiento:date, nombre:VARCHAR)

CP:dni

CNN:nombre, fechaNacimiento

**partido**(id:INT, estadio:VARCHAR, fecha:date, local:INT, visitante:INT)

CP:id

CF:local→Equipo(id), visitante→Equipo(id)

CNN:estadio, local, visitante

**pertenece**(dniJugador:INT, idEquipo:INT, fechaInicio:date)

CP:dniJugador, idEquipo

CF:dniJugador→Jugador(dni), idEquipo→Equipo(id)

CNN:fechaInicio

**juega**(dniJugador:INT, idPartido:INT, valorEstadística:INT, nombreEstadística:VARCHAR)

CP:dniJugador, idPartido, nombreEstadística

CF:dniJugador→Jugador(dni), nombreEstadistica→Estadistica(nombre), idPartido→Partido(id)

10)

CREATE TABLE **estadistica**(

nombre VARCHAR(40) PRIMARY KEY

);

CREATE TABLE **equipo**(

id INT PRIMARY KEY,

nombre VARCHAR(50) NOT NULL

);

CREATE TABLE **jugador**(

dni INT PRIMARY KEY,

fechaNacimiento DATE NOT NULL,

nombre VARCHAR(50) NOT NULL,

);

CREATE TABLE **partido**(

idPartido INT PRIMARY KEY,

local INT NOT NULL,

visitante INT NOT NULL,

fecha DATE,

estadio VARCHAR(50) NOT NULL,

FOREIGN KEY (local) REFERENCES equipo(id),

FOREIGN KEY (visitante) REFERENCES equipo(id)

);

CREATE TABLE **pertenece**(

dniJugador INT,

idEquipo INT,

fechaInicio DATE NOT NULL,

PRIMARY KEY (dniJugador, idEquipo)

FOREIGN KEY (dniJugador) REFERENCES jugador(dni),

FOREIGN KEY (idEquipo) REFERENCES equipo(id)

);

CREATE TABLE **juega**(

valorEstadistica INT,

dniJugador INT,

idPartido INT,

nombreEstadistica VARCHAR(40),

PRIMARY KEY (dniJugador, idPartido, nombreEstadistica),

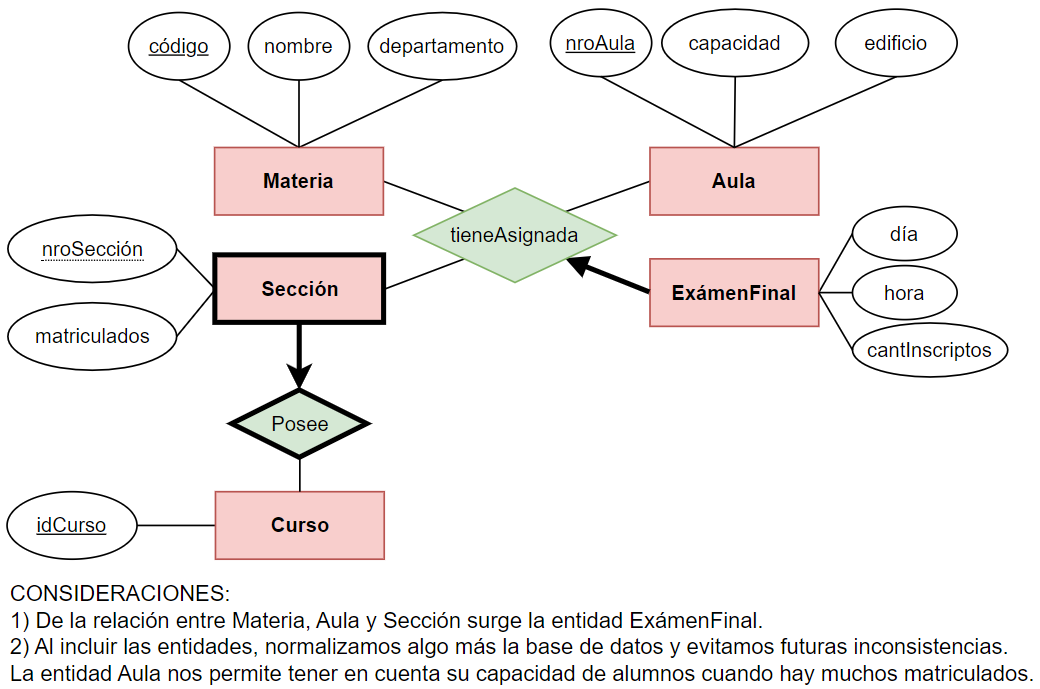
FOREIGN KEY (dniJugador) REFERENCES jugador(dni),

FOREIGN KEY (idPartido) REFERENCES partido(idP),

FOREIGN KEY (nombreEstadistica) REFERENCES estadistica(nombre)

)

7.



9)

**aula**(nroAula:INT, capacidad:INT, edificio:INT)

CP:nroAula

**materia**(codigo:INT, departamento:INT, nombre:VARCHAR)

CP:codigo

**seccion**(nroSeccion:INT, matriculados:INT, idC:INT)

CP:nroSeccion

CF:idC→Curso(idCurso)

CNN: idC

**curso**(idCurso:INT)

CP:idCurso

**examenFinal**(seccion:INT, materia:INT, idA:INT, dia:date, hora:VARCHAR, cantInscriptos:INT, idCurso:INT)

CP:idExamen

CF:seccion→Seccion(nroSeccion),

materia→Materia(codigo), idA→Aula(nroAula), idCurso→Curso(idCurso)

CNN:sección, materia, idA, idcurso

10)

CREATE TABLE **aula**(

nroAula INT PRIMARY KEY,

capacidad INT,

edificio INT

);

CREATE TABLE **materia**(

codigo INT PRIMARY KEY,

departamento INT,

nombre VARCHAR(40)

);

CREATE TABLE **curso**(

idCurso INT PRIMARY KEY

);

CREATE TABLE **seccion**(

nroSeccion INT PRIMARY KEY,

idC INT NOT NULL,

matriculados INT,

FOREIGN KEY (idC) REFERENCES curso(codigo) ON DELETE CASCADE

);

CREATE TABLE **examenFinal** (

idExamen INT PRIMARY KEY,

seccion INT NOT NULL,

materia INT NOT NULL,

idA INT NOT NULL,

dia date,

hora VARCHAR(5),

cantInscriptos INT,

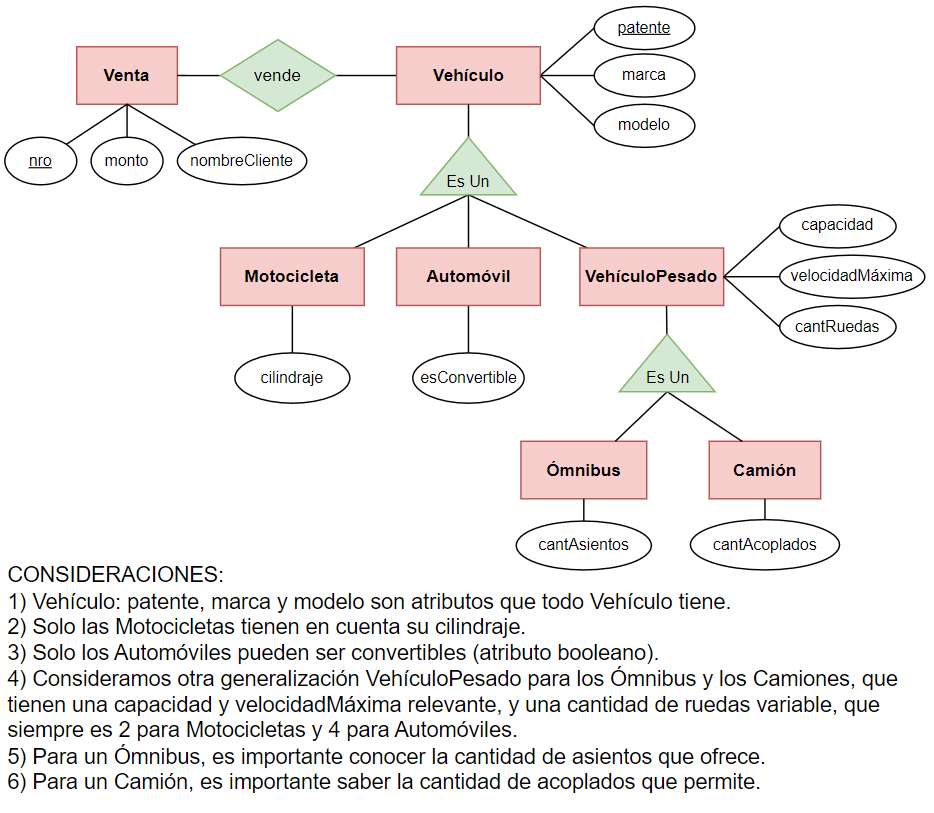
FOREIGN KEY (seccion) REFERENCES seccion(nroSeccion),

FOREIGN KEY (materia) REFERENCES materia(codigo),

FOREIGN KEY (materia) REFERENCES aula(nroAula)

)

8.

9)

**venta**(nro:INT, monto:INT, nombreCliente:VARCHAR)

CP:nro

CNN:monto, nombreCliente

**vende**(nro:INT, patente:VARCHAR)

CP:nro, patente

CF:nro→Venta(nro), patente→Vehiculo(patente)

Nota: en la entidad vehículo nos dimos cuenta que el atributo clave "patente" debería ser "id", pero decidimos respetar el modelo dado.

**vehiculo**(patente:VARCHAR, marca:VARCHAR, modelo:VARCHAR)

CP:patente

**moto**(patente:VARCHAR, cilindraje:INT)

CP:patente

CF:patente→Vehiculo(patente)

CNN:cilindraje

**auto**(patente:VARCHAR, esConvertible:BOOLEAN)

CP:patente

CF:patente→Vehiculo(patente)

CNN:esConvertible

**omnibus**(patente:INT, cantAsientos:INT, capacidad:INT, velMaax:INT, cantRuedas:INT)

CP:patente

CF:patente→Vehiculo(patente)

**camión**(patente:INT, cantAcoplados:INT, capacidad:INT, velMaax:INT, cantRuedas:INT)

CP:patente

CF:patente→Vehiculo(patente)

10)

CREATE TABLE **venta**(

nro INT PRIMARY KEY,

monto INT,

nombreCliente VARCHAR(50)

);

CREATE TABLE **vende**(

nro INT PRIMARY KEY,

patente INT NOT NULL,

FOREIGN KEY (nro) REFERENCES ventas(nro),

FOREIGN KEY (patente) REFERENCES vehiculo(patente)

);

CREATE TABLE **vehiculo**(

patente VARCHAR(7) PRIMARY KEY,

marca VARCHAR,

modelo VARCHAR

);

CREATE TABLE **auto** (

esConvertible INT NOT NULL,

patente INT PRIMARY KEY,

FOREIGN KEY (patente) REFERENCES vehiculo(patente),

);

CREATE TABLE **moto**(

cilindraje INT NOT NULL,

patente INT NOT NULL REFERENCES vehiculo(patente),

PRIMARY KEY(patente)

);

CREATE TABLE **omnibus**(

cantAsientos INT,

capacidad INT,

velMax INT,

cantRuedas INT,

patente INT NOT NULL,

FOREIGN KEY (patente) REFERENCES vehiculo(patente),

PRIMARY KEY(patente)

);

CREATE TABLE **camión**(

cantAcoplados INT,

capacidad INT,

velMax INT,

cantRuedas INT,

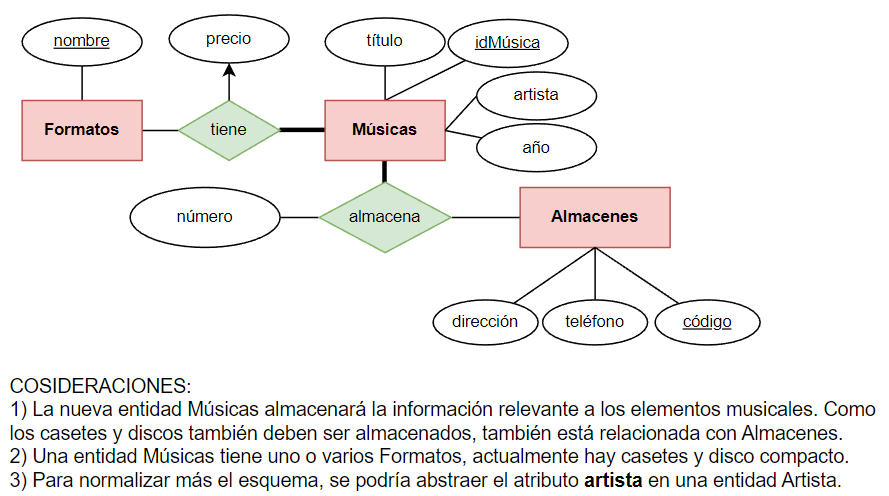
patente INT NOT NULL,

FOREIGN KEY (patente) REFERENCES vehiculo(patente),

PRIMARY KEY(patente)

);

9. b)



9)

**almacenes**(dirección:char, teléfono:INT, código:INT)

CP:código

**formato**(nombre:char)

CP:nombre

**músicas**(titulo:char, idMúsica:INT, artista:chat, año:INT)

CP:idMúsica

**tiene**(nombre:char, idMúsica:INT, precio:real)

CP:idMúsica, nombre

CF:idMúsica→Músicas(idMúsica), nombre→Formatos(nombre)

**almacena**(nombre:char, número:INT, código:INT, idMúsica:INT)

CP:código, idMúsica

CF:código→Almacenes(código), idMúsica→Músicas(idMúsica)

10)

CREATE TABLE **formatos**(

nombre VARCHAR(50) PRIMARY KEY

);

CREATE TABLE **almacenes**(

codigo INT PRIMARY KEY,

direccion VARCHAR(50),

telefono INT,

);

CREATE TABLE **musicas**(

idMusica INT PRIMARY KEY,

titulo VARCHAR(50),

artista VARCHAR(50),

anio INT

);

CREATE TABLE **tiene**(

precio FLOAT,

nombre VARCHAR(50),

FOREIGN KEY (nombre) REFERENCES formatos(nombre),

idMusica INT,

FOREIGN KEY (idMusica) REFERENCES musicas(idMusica)

PRIMARY KEY(idMusica, nombre)

);

CREATE TABLE **almacena**(

nombre VARCHAR(50),

codigo INT,

idMusica INT,

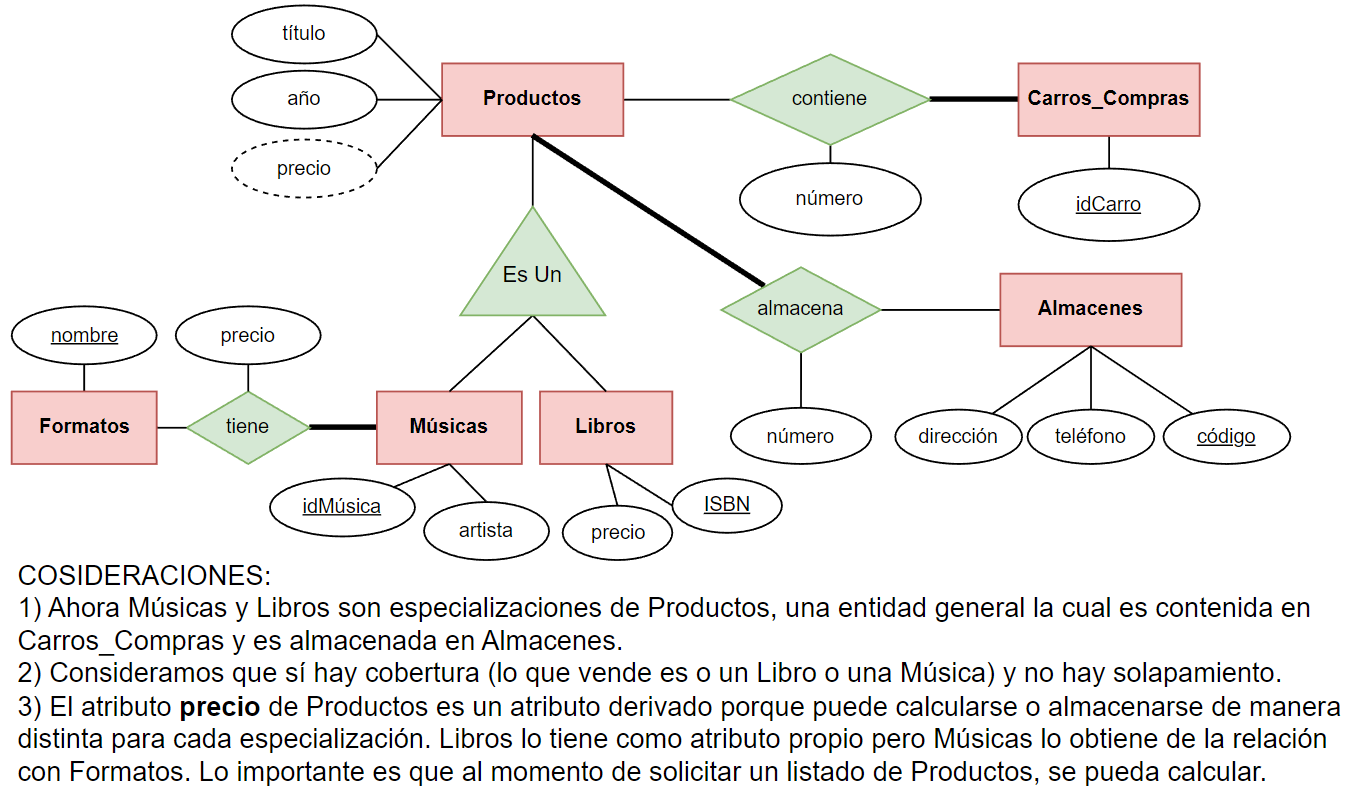
FOREIGN KEY codigo REFERENCES Almacenes(codigo),

FOREIGN KEY idMusica REFERENCES Músicas(idMúsica),

PRIMARY KEY(codigo, idMusica)

);

c)



9)

Aclaración: En la entidad Productos, se agrega un atributo clave “id” como clave primaria que es necesaria para, en el modelo conceptual, poder vincular las relaciones “músicas” y “libros”.

**productos**(id:INT, titulo:char, año:INT, precio:real, id:INT)

CP:id

**carro\_compras**(idCarro:INT)

CP:idCarro

**músicas**(id:INT, artista:char)

CP:id

CF:id → Productos(id)

CNN: artista

**libros**(id: INT, ISBN:INT, precio:real)

CP:id

CF: id → Productos(id)

CNN: ISBN

**formatos**(nombre:char)

CP:nombre

**almacenes**(dirección:char, teléfono:INT, código:INT)

CP:código

**contiene**(número:INT, idCarro:INT, idProducto:INT )

CP:idCarro, idProducto

CF: idCarro→CarrosCompras(idCarro)

idProducto→Productos(id)

**tiene**(nombreFormato:char, precio:real,idMúsica:INT)

CP:nombre, id

CF: nombre→Formatos(nombre)

id→Músicas(id)

**almacena**(idProducto:INT, número:INT, códigoAlmacen:INT)

CP:id, código

CF: idProducto → Productos(id)

códigoAlmacen → Almacenes(código)

10):

CREATE TABLE **productos**(

titulo VARCHAR(50),

año INT,

precio FLOAT,

id INT PRIMARY KEY

);

CREATE TABLE **carro\_compras**(

idCarro INT PRIMARY KEY

);

CREATE TABLE **músicas**(

id INT PRIMARY KEY,

FOREIGN KEY (id) REFERENCES Productos(id),

artista VARCHAR(30) NOT NULL

);

CREATE TABLE **libros**(

id INT PRIMARY KEY,

precio FLOAT,

ISB INT NOT NULL

);

CREATE TABLE **formatos**(

nombre VARCHAR(50) PRIMARY KEY

);

CREATE TABLE **almacenes**(

código INT PRIMARY KEY,

dirección VARCHAR(30),

teléfono INT

);

CREATE TABLE **contiene**(

numero VARCHAR(30),

idCarro INT,

idProducto INT,

FOREIGN KEY (idCarro) REFERENCES Carros Compras(idCarro),

FOREIGN KEY (idProducto) REFERENCES Productos(id),

PRIMARY KEY(idCarro, idProducto)

);

CREATE TABLE **tiene**(

nombreFormato VARCHAR(30),

idMusica INT,

precio FLOAT,

FOREIGN KEY nombreFormato REFERENCES formatos(nombre),

FOREIGN KEY idMusica REFERENCES musicas(id),

PRIMARY KEY (idMusica, nombreFormato)

);

CREATE TABLE **almacena**(

idProducto INT,

numero INT,

codigoAlmacen INT,

FOREIGN KEY idProducto REFERENCES Producto(id),

FOREIGN KEY codigoAlmacen REFERENCES Almacen(codigo),

PRIMARY KEY (idProducto, codigoAlmacen)

);