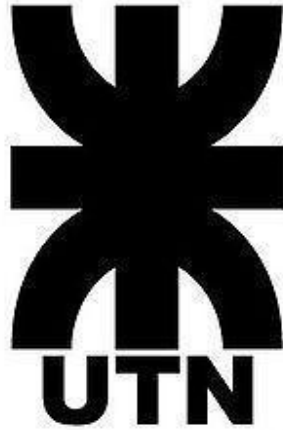


BASE DE DATOS



Integrantes

ALBORNOZ, Maria Leonor

COCITO, Maximiliano Hernán

KINWEILER, Victor Fabricio

MOSELLI, Yamil Apas

SCHEFER, Mauricio Nicolás.

VELAZCO GEZ SCHEGTEL, Juan Ignacio

2024

GRUPO 8

Entiendo, aquí está el texto con formato simple que puedes copiar y pegar directamente en un documento de Word:

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.
 - El esquema de relación es uno de los componentes de una relación en el modelo relacional. Dicho esquema hace referencia a una descripción de las cabezas que componen las tablas, también conocidas como instancias de relación. Esta descripción incluye el nombre de la relación, el nombre de cada atributo y el dominio de dichos atributos. Ej: Alumnos (idd: String, nombre: string, etc...)
 - El esquema de una base de datos relacional es un conjunto de esquemas de las relaciones de base de datos.
 - Una instancia de una relación es un conjunto de tuplas o registros, en el que cada tupla tiene el mismo número de campos que el esquema de la relación.
 - La cardinalidad de una relación hace referencia a la cantidad de tuplas que tiene una instancia.
 - El grado de una relación es el número de campos o columnas.
2. ¿Cuántas tuplas distintas hay en una instancia de una relación de cardinalidad 22?
 - En una instancia de cardinalidad 22 hay 22 tuplas porque no puede haber 2 tuplas idénticas.
3. ¿Provee el modelo relacional independencia física y lógica de datos?
 - Sí, provee independencia física y lógica de datos, esto lo hace para ser lo más transparente posible con el cliente/usuario.
4. ¿Cuál es la diferencia entre una clave candidata y la clave primaria de una relación? ¿Qué es una superclave?
 - La clave candidata es un subconjunto de la superclave que me permiten identificar unívocamente la relación.
 - La clave primaria de una relación es el subconjunto más pequeño de las claves candidatas que me permite identificar unívocamente al registro.
 - La superclave es el conjunto de campos que integran la relación.
5. Considere la instancia de la tabla Alumnos mostrada en la siguiente figura: 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?
 - Un conjunto de atributos que NO son clave candidata son: Nombre y Edad
 - Un conjunto de atributos que son clave candidata son: Nombre y Login

6. ¿Qué es una restricción de clave foránea? ¿Por qué son importantes tales restricciones? ¿Qué es integridad referencial?
- Conjunto de campos en una relación que se usa para “referenciar” a una tupla en otra relación. (Debe corresponder con una clave primaria de la segunda relación.)
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)
 - Liste todas las claves foráneas de estas relaciones.
 - Claves Foráneas de Matriculados:
 - ide → Alumnos(ide)
 - ida → Asignaturas(ida)
 - Claves Foráneas de Imparte:
 - idp → Profesores(idp)
 - ida → Asignaturas(ida)
 - Claves Foráneas de Impartida_en:
 - ida → Asignaturas(ida)
 - nau → Aulas(nau)
 - 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.
 - La relación imparte podríamos tener una restricción de participación, nos indicará que todos los profesores deben impartir al menos una clase.
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)
 - Dé un ejemplo de clave foránea.
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.**

- 1) **automoviles** (patente: CHAR(10), modelo: CHAR(20), marca: CHAR(30), nrochasis: INTEGER, nroacci: INTEGER, dni: INTEGER)

CP: (patente)

CF: dni -> clientes(dni)

CNN: dni

clientes (dni: INTEGER, nombre_y_apellido: CHAR(40), tel: INTEGER, email: CHAR(40))

CP: (dni)

CF: -

CNN: -

- 2) **pacientes** (dni: INTEGER, nya: CHAR(30), tel: INTEGER, direc: CHAR(30))

CP: (dni)

CF: -

CNN: -

medicos (dni: INTEGER, nro_mat: CHAR(30), nombre_y_apellido: CHAR(40), especialidad: CHAR(20))

CP: (dni)

CF: -

CNN: -

examenes_medicos (nro_exam: INTEGER, fecha_exam: DATE, tipo_exam: CHAR(30), dni: INTEGER, nro_mat: CHAR(30))

CP: (nro_exam)

CF: dni -> pacientes(dni), nro_mat -> medicos(nro_mat)

CNN: dni, nro_mat

- 3) **materias** (id_mat: INTEGER, programa: CHAR(200), titulo: CHAR(200))

CP: (id_mat)

CF: -

CNN: -

requiere (idmat_requerida: CHAR(30), idmat_requirente: CHAR(30))

CP: (idmat_requerida, idmat_requirente)

CF: idmat_requerida -> materias(id_mat), idmat_requirente -> materias(id_mat)

CNN: -

dictado_de_materias (anio: INTEGER, semestre: CHAR(10), nro_secc: CHAR(10), horarios: CHAR(20), aulas: CHAR(10), id_mat: INTEGER)

CP: (id_mat, anio, semestre, nro_secc)

CF: id_mat -> materias(id_mat) EN BORRADO CASCADA EN ACTUALIZACIÓN CASCADA

CNN: -

profesores (titulo: CHAR(50), depto: CHAR(30), nomyape: CHAR(30), nro_ident: INTEGER)

CP: (nro_ident)

CF: -

CNN: -

estudiantes (dni: INTEGER, nombre_y_apellido: CHAR(40), nro_legajo: INTEGER, tel: INTEGER, email: CHAR(50))

CP: (dni)

CF: -

CNN: -

realiza (nro_ident: INTEGER, anio: INTEGER, semestre: CHAR(10), nro_secc: CHAR(10), id_mat: INTEGER)

CP: (id_mat, nro_ident, anio, semestre, nro_secc)

CF: id_mat -> materias(id_mat), anio -> dictado_de_materias(anio), nro_secc -> dictado_de_materias(nro_secc), semestre -> dictado_de_materias(semestre), nro_ident -> profesores(nro_ident)

CNN: -

asiste (nro_mat: INTEGER, nota: REAL, dni: INTEGER, anio: INTEGER, semestre: CHAR(10), nro_secc: CHAR(10))

CP: (dni, anio, semestre, nro_secc)

CF: dni -> estudiantes(dni), anio -> dictado_de_materias(anio), semestre -> dictado_de_materias(semestre), nro_secc -> dictado_de_materias(nro_secc)

CNN: -

4.a

materia (id_mat: INTEGER, nombre: CHAR(30), nivel: CHAR(2))

CP: (id_mat)

CF: -

CNN: -

dictado (aula: CHAR(2), horario: CHAR(10), anio: INTEGER, semestre: CHAR(10), nro_secc: CHAR(10), id_mat: INTEGER)

CP: (id_mat, anio, semestre, nro_secc)

CF: id_mat -> materia(id_mat)

EN BORRADO CASCADA EN ACTUALIZACIÓN CASCADA

CNN: -

estudiante (dni: INTEGER, nombre_y_apellido: CHAR(40), nro_legajo: INTEGER, email: CHAR(40), fecha_nacimiento: DATE)

CP: (dni)

CF: -

CNN: -

examen (fecha: DATE, nota: REAL, tipo: CHAR(20), id_exam: INTEGER, dni: INTEGER, anio: INTEGER, semestre: CHAR(10), nro_secc: CHAR(10))

CP: (id_exam)

CF: dni -> estudiante(dni), anio -> dictado(anio), semestre -> dictado(semestre), nro_secc -> dictado(nro_secc)

CNN: dni, anio, semestre, nro_secc

4.b

materia (id_mat: INTEGER, nombre: CHAR(30), nivel: CHAR(20))

CP: (id_mat)

CF: -

CNN: -

dictado (aula: CHAR(30), horario: CHAR(20), nro_secc: INTEGER, semestre: INTEGER, anio: INTEGER, id_mat: INTEGER)

CP: (nro_secc, semestre, anio, id_mat)

CF: id_mat -> materia(id_mat)

EN BORRADO CASCADA EN ACTUALIZACIÓN CASCADA

CNN: -

examen (id_exam: INTEGER, tipo: CHAR(30), fecha: DATE, nota: INTEGER)

CP: (id_exam)

CF: -

CNN: -

estudiante (dni: INTEGER, NyA: CHAR(30), nro_lega: INTEGER, email: CHAR(20), fecha_na: DATE)

CP: (dni)

CF: -

CNN: fecha_na

posee (dni: INTEGER, anio: INTEGER, semestre: INTEGER, nro_secc: INTEGER)

CP: (dni, anio, semestre, nro_secc)

CF: dni -> estudiante(dni), anio -> dictado(anio), semestre -> dictado(semestre), nro_secc -> dictado(nro_secc)

CNN: -

tiene (id_exam: INTEGER, dni: INTEGER, anio: INTEGER, semestre: INTEGER, nro_secc: INTEGER)

CP: (id_exam, dni, anio, semestre, nro_secc)

CF: id_exam -> examen(id_exam), dni -> posee(dni), anio -> posee(anio), semestre -> posee(semestre), nro_secc -> posee(nro_secc)

CNN: -

5.

jugador (id_jugador: INTEGER, nombre_y_apellido: CHAR(30), dorsal: INTEGER, posicion: INTEGER, faltas_totales: INTEGER, goles_totales: INTEGER, minutos_totales: INTEGER, asistencias_totales: INTEGER)

CP: (id_jugador)

CF: -

CNN: -

partido (rival: CHAR(30), id_partido: INTEGER, duracion: INTEGER, fecha: DATE, hora: INTEGER, estadio: CHAR(30), resultado: INTEGER)

CP: (id_partido)

CF: -

CNN: -

participa (minutos_jugador: INTEGER, faltas: INTEGER, goles: INTEGER, asistencias: INTEGER, id_jugador: INTEGER, id_partido: INTEGER)

CP: (id_jugador, id_partido)

CF: id_jugador -> jugador(id_jugador), id_partido -> partido(id_partido)

CNN: -

6.

equipo (id_equipo: INTEGER, nombre: CHAR(CHAR), puntos: INTEGER)

CP: (id_equipo)

CF: -

CNN: -

juega_contra (id_equipo_local: INTEGER, id_equipo_visitante: INTEGER, tiros_al_arco_local: INTEGER, tiros_al_arco_visitante: INTEGER, duracion: INTEGER, goles_local: INTEGER, goles_visitante: INTEGER, pasos_dados_local: INTEGER, pasos_dados_visitante: INTEGER, posesion_local: INTEGER, posesion_visitante: INTEGER)

CP: (id_equipo_local, id_equipo_visitante)

CF: id_equipo_local -> equipo(id_equipo), id_equipo_visitante -> equipo(id_equipo)

CNN: -

participa (goles: INTEGER, faltas: INTEGER, minutos_jugados: INTEGER, asistencias: INTEGER, id_equipo_local: INTEGER, id_equipo_visitante: INTEGER, id_jugador: INTEGER)

CP: (id_jugador, id_equipo_local, id_equipo_visitante)

CF: id_jugador -> jugador(id_jugador), id_equipo_local -> equipo(id_equipo), id_equipo_visitante -> equipo(id_equipo)

CNN: -

jugador (asistencias_totales: INTEGER, minutos_totales: INTEGER, faltas_totales: INTEGER, goles_totales: INTEGER, dorsal: INTEGER, nombre_y_apellido: CHAR(30), posicion: INTEGER, id_jugador: INTEGER, id_equipo: INTEGER)

CP: (id_jugador)

CF: id_equipo -> equipo(id_equipo)

CNN: -

7.

comision (id_materia: INTEGER, id_comision: INTEGER, matriculados: INTEGER)

CP: (id_materia, id_comision)

CF: id_materia -> materia(id_comision) EN BORRADO CASCADA

CNN: -

materia (anio_plan: INTEGER, nombre: CHAR(30), departamento: CHAR(30), id_materia: INTEGER)

CP: (id_materia)

CF: -

CNN: -

examen (id_materia: INTEGER, id_aula: INTEGER, id_examen: INTEGER, nombre_alumno: CHAR(30), hora: INTEGER, fecha: DATE, calificacion: INTEGER, profesor_titular: CHAR(30))

CP: (id_examen)

CF: id_aula -> aula(id_aula), id_materia -> materia(id_materia)

CNN: id_materia, id_aula

aula (id_aula: INTEGER, capacidad: INTEGER, edificio: INTEGER)

CP: (id_aula)

CF: -

CNN: -

8.

vendedor (id_vendedor: INTEGER)

CP: (id_vendedor)

CF: -

CNN: -

vehiculo (marca: CHAR(30), anio: INTEGER, cilindrada: REAL, modelo: CHAR(30), color: CHAR(20), kilometros: REAL, transmision: CHAR(20), id_vehiculo: INTEGER, id_vendedor: INTEGER)

CP: (id_vehiculo)

CF: id_vendedor -> vendedor(id_vendedor)

CNN: -

motocicleta (transmision_a_rueda: INTEGER, tipo: CHAR(30), carenado: CHAR(20), id_vehiculo: INTEGER)

CP: (id_vehiculo)

CF: id_vehiculo -> vehiculo(id_vehiculo) EN BORRADO CASCADA

CNN: -

automovil (id_vehiculo: INTEGER, segmento: CHAR(30), carroceria: CHAR(30))

CP: (id_vehiculo)

CF: id_vehiculo -> vehiculo(id_vehiculo) EN BORRADO CASCADA

CNN: -

camiones(id_vehiculo: INTEGER, nro_de_ejes: INTEGER, disposicion_del_motor: CHAR(30), capacidad_del_remolque: INTEGER)

CP: (id_vehiculo)

CF: id_vehiculo -> vehiculo(id_vehiculo) EN BORRADO CASCADA

CNN: -

omnibus (id_vehiculo: INTEGER, nro_de_asientos: INTEGER, marca_de_la_carroceria: CHAR(30))

CP: (id_vehiculo)

CF: id_vehiculo -> vehiculo(id_vehiculo) EN BORRADO CASCADA

CNN: -

10.b

autores (nombre_autor: CHAR(30), direccion: CHAR(30), url: CHAR(30))

CP: (nombre, direccion)

CF: -

CNN: -

editores (nombre_editor: CHAR(30), direcc: CHAR(30), telefono: INTEGER, url: CHAR(30))

CP: (nombre)

CF: -

CNN: -

libros(isbn: CHAR(30), anio: INTEGER, precio: INTEGER, titulo: CHAR(30), nombre_editor: CHAR(30))

CP: (isbn, nombre_editor)

CF: nombre_editor -> editores(nombre_editor)

CNN: nombre_editor

almacenes (codigo: CHAR(30), telefono: INTEGER, direccion: CHAR(30))

CP: (codigo)

CF: -

CNN: -

clientes(direccion_email: CHAR(30), nombre: CHAR(30), direccion_clientes: CHAR(30), telefono_clientes: INTEGER)

CP: (direccion_email)

CF: -

CNN: -

carro_compras (id_carro: INTEGER, direccion_email: CHAR(30))

CP: (id_carro, direccion_email)

CF: direccion_email -> clientes(direccion_email)

CNN: direccion_email

albums (id_album: INTEGER, genero: CHAR(40), precio: INTEGER, artista: CHAR(30), anio: INTEGER, nombre_album: CHAR(30), discografia: CHAR(30), duracion: INTEGER)

CP: (id_album)

CF: -

CNN: -

cds (id_album: INTEGER, dvd_vivo: CHAR(30))

CP: (id_album)

CF: id_album -> albums(id_album) EN BORRADO CASCADA

CNN: -

cassettes (id_album: INTEGER, id_cassettes: INTEGER)

CP: (id_album)

CF: id_album -> albums(id_album) EN BORRADO CASCADA

CNN: -

10.c

autores (nombre: CHAR(30), direccion: CHAR(30), url: CHAR(100))

CP: (nombre, dirección)

CF: -

CNN: -

producto (id_producto: INTEGER, genero: CHAR(30), titulo: CHAR(80), precio: REAL, anio: INTEGER)

CP: (id_producto)

CF: -

CNN: -

albums (id_producto: INTEGER, discografica: CHAR(40), duracion: INTEGER, artista: CHAR(40))

CP: (id_producto)

CF: id_producto -> producto(id_producto) EN BORRADO CASCADA

CNN: -

cassettes (id_producto: INTEGER)

CP: (id_producto)

CF: id_producto -> albums(id_producto) EN BORRADO CASCADA

CNN: -

cds (id_producto: INTEGER, dvd_vivo: CHAR(2))

CP: (id_producto)

CF: id_producto -> albums(id_producto) EN BORRADO CASCADA

CNN: -

libros (id_producto: INTEGER, isbn: CHAR(11), nombre: CHAR(30))

CP: (id_producto)

CF: id_producto -> producto(id_producto) EN BORRADO CASCADA, nombre -> editores(nombre)

CNN: nombre

editores (nombre: CHAR(30), direccion: CHAR(30), telefono: INTEGER, url: CHAR(80))

CP: (nombre)

CF: -

CNN: -

almacenes (codigo: INTEGER, telefono: INTEGER, direccion: CHAR(40))

CP: (codigo)

CF: -

CNN: -

almacena (numero: INTEGER, id_producto: INTEGER, codigo: INTEGER)

CP: (id_producto, codigo)

CF: id_producto -> producto(id_producto), codigo -> almacenes(codigo)

CNN: -

clientes (nombre: CHAR(30), direccion: CHAR(30), telefono: INTEGER, direccion_email: CHAR(50))

CP: (direccion_email)

CF: -

CNN: -

carro_compras (id_carro: INTEGER, direccion_email: CHAR(30))

CP: (id_carro)

CF: direccion_email -> clientes(direccion_email)

CNN: direccion_email

contiene (numero: INTEGER, id_carro: INTEGER, id_producto: INTEGER)

CP: (id_carro, id_producto)

CF: id_carro -> carro_compras(id_carro), id_producto -> producto(id_producto)

CNN: -

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.

```
CREATE TABLE automoviles (  
    patente CHAR(10),  
    modelo CHAR(20),  
    marca CHAR(30),  
    nrochasis INTEGER,  
    nroacci INTEGER,  
    dni INTEGER NOT NULL,  
    PRIMARY KEY (patente),  
    FOREIGN KEY (dni) REFERENCES clientes(dni),  
)
```

```
CREATE TABLE clientes (  
    dni INTEGER,  
    nombre_y_apellido CHAR(40),  
    tel INTEGER,  
    email CHAR(40),  
    PRIMARY KEY (dni)
```


)

2.

CREATE TABLE **pacientes** (

dni INTEGER,

nya CHAR (30),

tel INTEGER,

direc CHAR (30),

PRIMARY KEY (dni),

)

CREATE TABLE **medicos** (

dni INTEGER,

nro_mat CHAR(30),

nombre_y_apellido CHAR(40),

especialidad CHAR(20),

PRIMARY KEY (dni)

)

CREATE TABLE **exámenes_medicos** (

nro_exam INTEGER,

fecha_exam DATE,

tipo_exam CHAR(30),

dni INTEGER NOT NULL,

nro_mat CHAR(30) NOT NULL,

```
PRIMARY KEY (nro_exam),  
FOREIGN KEY (dni) REFERENCES pacientes(dni),  
FOREIGN KEY (nro_mat) REFERENCES medicos(nro_mat)  
)
```

3.

```
CREATE TABLE materias (  
    id_mat INTEGER,  
    programa CHAR(200),  
    titulo CHAR(200),  
    PRIMARY KEY (id_mat)  
)
```

```
CREATE TABLE requiere (  
    idmat_requerida CHAR(30),  
    idmat_requirente CHAR (30),  
    PRIMARY KEY (idmat_requerida, idmat_requirente),  
    FOREIGN KEY (idmat_requerida) REFERENCES materias (id_mat),  
    FOREIGN KEY (idmat_requirente) REFERENCES materias (id_mat),  
)
```

```
CREATE TABLE dictado_de_materias (  
    anio INTEGER,  
    semestre CHAR(10),  
    nro_secc CHAR(10),  
    horarios CHAR(20),
```

```
aulas CHAR(10),  
  
id_mat INTEGER,  
  
PRIMARY KEY (id_mat, anio, semestre, nro_secc),  
  
FOREIGN KEY (id_mat) REFERENCES materias(id_mat) ON DELETE CASCADE ON UPDATE  
CASCADE  
)
```

```
CREATE TABLE profesores (  
  
    titulo CHAR (50),  
  
    depto CHAR (30),  
  
    nomyape CHAR (30),  
  
    nro_ident INTEGER,  
  
    PRIMARY KEY (nro_ident),  
  
)
```

```
CREATE TABLE estudiantes (  
  
    dni INTEGER,  
  
    nombre_y_apellido CHAR(40),  
  
    nro_legajo INTEGER,  
  
    tel INTEGER,  
  
    email CHAR(50),  
  
    PRIMARY KEY (dni)  
)
```

```
CREATE TABLE realiza (  
  
    nro_ident INTEGER,  
  
    anio INTEGER,
```

```

semestre CHAR (10),
nro_secc CHAR(10),
id_mat INTEGER,
PRIMARY KEY (id_mat, nro_ident, anio, semestre, nro_secc),
FOREIGN KEY (id_mat) REFERENCES materias(id_mat),
FOREIGN KEY (anio) REFERENCES dictado_de_materias(anio),
FOREIGN KEY (nro_secc) REFERENCES dictado_de_materias(nro_secc),
FOREIGN KEY (semestre) REFERENCES dictado_de_materias(semestre),
FOREIGN KEY (nro_ident) REFERENCES profesores(nro_ident)
)

```

```

CREATE TABLE asiste (
    nro_mat INTEGER,
    nota REAL,
    dni INTEGER,
    anio INTEGER,
    semestre CHAR(10),
    nro_secc CHAR(10),
    PRIMARY KEY (dni, anio, semestre, nro_secc),
    FOREIGN KEY (dni) REFERENCES estudiantes(dni),
    FOREIGN KEY (anio) REFERENCES dictado_de_materias(anio),
    FOREIGN KEY (semestre) REFERENCES dictado_de_materias(semestre),
    FOREIGN KEY (nro_secc) REFERENCES dictado_de_materias(nro_secc)
)

```

```
CREATE TABLE materia (  
    id_mat INTEGER,  
    nombre CHAR(30),  
    nivel CHAR(2),  
    PRIMARY KEY (id_mat)  
)
```

```
CREATE TABLE dictado (  
    aula CHAR(2),  
    horario CHAR(10),  
    anio INTEGER,  
    semestre CHAR(10),  
    nro_secc CHAR(10),  
    id_mat INTEGER,  
    PRIMARY KEY (id_mat, anio, semestre, nro_secc),  
    FOREIGN KEY (id_mat) REFERENCES materia(id_mat) ON DELETE CASCADE ON UPDATE  
    CASCADE  
)
```

```
CREATE TABLE estudiante (  
    dni INTEGER,  
    nombre_y_apellido CHAR(40),  
    nro_legajo INTEGER,  
    email CHAR(40),  
    fecha_nacimiento DATE,  
    PRIMARY KEY (dni)
```

)

```
CREATE TABLE examen (  
    fecha DATE,  
    nota REAL,  
    tipo CHAR (20),  
    id_exam INTEGER,  
    dni INTEGER NOT NULL,  
    anio INTEGER NOT NULL,  
    semestre CHAR(10) NOT NULL,  
    nro_secc CHAR(10) NOT NULL,  
    PRIMARY KEY (id_exam),  
    FOREIGN KEY (dni) REFERENCES estudiante(dni),  
    FOREIGN KEY (anio) REFERENCES dictado(anio),  
    FOREIGN KEY (semestre) REFERENCES dictado(semestre),  
    FOREIGN KEY (nro_secc) REFERENCES dictado(nro_secc)  
)
```

4.b

```
CREATE TABLE materia (  
    id_mat INTEGER,  
    nombre CHAR(30),  
    nivel CHAR(20),  
    PRIMARY KEY (id_mat)  
)
```

```
CREATE TABLE dictado (  
    aula CHAR(30),  
    horario CHAR(20),  
    nro_secc INTEGER,  
    semestre INTEGER,  
    anio INTEGER,  
    id_mat INTEGER,  
    PRIMARY KEY (nro_secc, semestre, anio, id_mat),  
    FOREIGN KEY (id_mat) REFERENCES materia(id_mat) ON DELETE CASCADE ON UPDATE  
    CASCADE  
)
```

```
CREATE TABLE examen (  
    id_exam INTEGER,  
    tipo CHAR(30),  
    fecha DATE,  
    nota INTEGER,  
    PRIMARY KEY (idexam)  
)
```

```
CREATE TABLE estudiante (  
    dni INTEGER,  
    NyA CHAR(30),  
    nro_lega INTEGER,  
    email CHAR(20),  
    fecha_na DATE NOT NULL,
```

PRIMARY KEY (dni)
)

CREATE TABLE **posee** (
dni INTEGER,
anio INTEGER,
semestre INTEGER,
mro_secc INTEGER,
PRIMARY KEY (dni, anio, semestre, nro_secc),
FOREIGN KEY (dni) REFERENCES estudiante(dni),
FOREIGN KEY (anio) REFERENCES dictado (anio),
FOREIGN KEY (semestre) REFERENCES dictado (semestre),
FOREIGN KEY (nro_secc) REFERENCES dictado (nro_secc)
)

CREATE TABLE **tiene** (
id_exam INTEGER,
dni INTEGER,
anio INTEGER,
semestre INTEGER,
mro_secc INTEGER,
PRIMARY KEY (id_exam, dni, anio, semestre, nro_secc)
FOREIGN KEY (id_exam) REFERENCES examen(id_exam)
FOREIGN KEY (dni) REFERENCES posee(dni)
FOREIGN KEY (anio) REFERENCES posee(anio)

FOREIGN KEY (semestre) REFERENCES posee(semestre)

FOREIGN KEY (nro_secc) REFERENCES posee(nro_secc)

)

5.

CREATE TABLE **jugador** (

id_jugador INTEGER,

nombre_y_apellido CHAR(30),

dorsal INTEGER,

posicion INTEGER,

faltas_totales INTEGER,

goles_totales INTEGER,

minutos_totales INTEGER,

asistencias_totales INTEGER,

PRIMARY KEY (id_jugador),

)

CREATE TABLE **partido**(

rival CHAR(30),

id_partido INTEGER,

duracion INTEGER,

fecha DATE,

hora INTEGER,

estadio CHAR(30),

resultado INTEGER,

PRIMARY KEY (id_partido),

)

```
CREATE TABLE participa (  
    minutos_jugador INTEGER,  
    faltas INTEGER,  
    goles INTEGER,  
    asistencias INTEGER,  
    id_jugador INTEGER,  
    id_partido INTEGER,  
    PRIMARY KEY (id_jugador, id_partido),  
    FOREIGN KEY (id_jugador) REFERENCES jugador(id_jugador),  
    FOREIGN KEY (id_partido) REFERENCES partido(id_partido)  
)
```

6.

```
CREATE TABLE equipo (  
    id_equipo INTEGER,  
    nombre CHAR(CHAR),  
    puntos INTEGER,  
    PRIMARY KEY (id_equipo)  
)
```

```
CREATE TABLE juega_contra (  
    id_equipo_local INTEGER,  
    id_equipo_visitante INTEGER,  
    tiros_al_arco_local INTEGER,
```

```
    tiros_al_arco_visitante INTEGER,

    duracion INTEGER,

    goles_local INTEGER,

    goles_visitante INTEGER,

    pasos_dados_local INTEGER,

    pasos_dados_visitante INTEGER,

    posesion_local INTEGER,

    posesion_visitante INTEGER,

    PRIMARY KEY (id_equipo_local, id_equipo_visitante),

    FOREIGN KEY (id_equipo_local) REFERENCES equipo(id_equipo),

    FOREIGN KEY (id_equipo_visitante) REFERENCES equipo(id_equipo)

)
```

```
CREATE TABLE participa (

    goles INTEGER,

    faltas INTEGER,

    minutos_jugados INTEGER,

    asistencias INTEGER,

    id_equipo_local INTEGER,

    id_equipo_visitante INTEGER,

    id_jugador INTEGER,

    PRIMARY KEY (id_jugador, id_equipo_local, id_equipo_visitante),

    FOREIGN KEY (id_jugador) REFERENCES jugador(id_jugador),

    FOREIGN KEY (id_equipo_local) REFERENCES equipo(id_equipo_local),

    FOREIGN KEY (id_equipo_visitante) REFERENCES equipo(id_equipo_visitante),

)
```

)

```
CREATE TABLE jugador (  
    asistencias_totales INTEGER,  
    minutos_totales INTEGER,  
    faltas_totales INTEGER,  
    goles_totales INTEGER,  
    dorsal INTEGER,  
    nombre_y_apellido CHAR(30),  
    posicion INTEGER,  
    id_jugador INTEGER,  
    id_equipo INTEGER,  
    PRIMARY KEY (id_jugador),  
    FOREIGN KEY (id_equipo) REFERENCES equipo(id_equipo)  
)
```

7.

```
CREATE TABLE comision (  
    id_materia INTEGER,  
    id_comision INTEGER,  
    matriculados INTEGER,  
    PRIMARY KEY (id_materia, id_comision),  
    FOREIGN KEY (id_materia) REFERENCES materia(id_comision) ON DELETE CASCADE  
)
```

```
CREATE TABLE materia (
```

```
    anio_plan INTEGER,  
    nombre CHAR(30),  
    departamento CHAR(30),  
    id_materia INTEGER,  
    PRIMARY KEY (id_materia)  
)
```

```
CREATE TABLE examen (  
    id_materia INTEGER NOT NULL,  
    id_aula INTEGER NOT NULL,  
    id_examen INTEGER,  
    nombre_alumno CHAR(30),  
    hora INTEGER,  
    fecha DATE,  
    calificacion INTEGER,  
    profesor_titular CHAR(30),  
    PRIMARY KEY (id_examem),  
    FOREIGN KEY (id_aula) REFERENCES aula(id_aula),  
    FOREIGN KEY (id_materia) REFERENCES materia(id_materia)  
)
```

```
CREATE TABLE aula (  
    id_aula INTEGER,  
    capacidad INTEGER,  
    edificio INTEGER,  
    PRIMARY KEY (id_aula)
```

)

8.

```
CREATE TABLE vendedor (  
    id_vendedor INTEGER,  
    PRIMARY KEY (id_vendedor)  
)
```

```
CREATE TABLE vehiculo (  
    marca CHAR(30),  
    anio INTEGER,  
    cilindrada REAL,  
    modelo CHAR(30),  
    color CHAR(20),  
    kilometros REAL,  
    transmision CHAR(20),  
    id_vehiculo INTEGER,  
    id_vendedor INTEGER,  
    PRIMARY KEY (id_vehiculo),  
    FOREIGN KEY (id_vendedor) REFERENCES evendedor(id_vendedor)  
)
```

```
CREATE TABLE motocicleta (  
    transmision_a_rueda INTEGER,  
    tipo CHAR(30),  
    carenado CHAR(20),
```

```
id_vehiculo INTEGER,  
  
PRIMARY KEY (id_vehiculo),  
  
FOREIGN KEY (id_vehiculo) REFERENCES vehiculo(id_vehiculo) ON DELETE CASCADE  
)
```

```
CREATE TABLE automovil (  
  
id_vehiculo INTEGER,  
  
segmento CHAR(30),  
  
carroceria CHAR(30),  
  
PRIMARY KEY (id_vehiculo),  
  
FOREIGN KEY (id_vehiculo) REFERENCES vehiculo(id_vehiculo) ON DELETE CASCADE  
)
```

```
CREATE TABLE camiones (  
  
id_vehiculo INTEGER,  
  
nro_de_ejes INTEGER,  
  
disposicion_del_motor CHAR(30),  
  
capacidad_del_remolque INTEGER,  
  
PRIMARY KEY (id_vehiculo),  
  
FOREIGN KEY (id_vehiculo) REFERENCES vehiculo(id_vehiculo) ON DELETE CASCADE  
)
```

```
CREATE TABLE omnibus (  
  
id_vehiculo INTEGER,  
  
nro_de_asientos INTEGER,  
  
marca_de_la_carroceria CHAR(30),
```

```
PRIMARY KEY (id_vehiculo),  
FOREIGN KEY (id_vehiculo) REFERENCES vehiculo(id_vehiculo) ON DELETE CASCADE  
)
```

10.b

```
CREATE TABLE autores(  
    nombre_autor CHAR(30),  
    direccion CHAR(30),  
    url CHAR(30),  
    PRIMARY KEY (nombre, direccion),  
)
```

```
CREATE TABLE editores (  
    nombre_editor CHAR(30)  
    direcc CHAR(30),  
    telefono INTEGER,  
    url CHAR(30),  
    PRIMARY KEY (nomb),  
)
```

```
CREATE TABLE libros (  
    isbn CHAR(30),  
    anio INTEGER,  
    precio INTEGER,  
    titulo CHAR(30),  
    nombre_editor CHAR(30) NOT NULL,
```



```
PRIMARY KEY (isbn, nombre_editor),  
FOREIGN KEY (nombre_editor) REFERENCES editores(nombre_editor),  
)
```

```
CREATE TABLE almacenes (  
    codigo CHAR(30),  
    telefono INTEGER,  
    direccion CHAR(30),  
    PRIMARY KEY (codigo),  
)
```

```
CREATE TABLE clientes (  
    direccion_email CHAR(30),  
    nombre CHAR(30),  
    direccion_clientes CHAR(30),  
    telefono_clientes INTEGER,  
    PRIMARY KEY (direccion_email),  
)
```

```
CREATE TABLE carro_compras (  
    id_carro INTEGER,  
    direccion_email CHAR(30) NOT NULL,  
    PRIMARY KEY (id_carro, direccion_email),  
    FOREIGN KEY (direccion_email) REFERENCES clientes(direccion_email),  
)
```

```
CREATE TABLE albums (  
    id_album INTEGER,  
    genero CHAR(40),  
    precio INTEGER,  
    artista CHAR(30),  
    anio INTEGER,  
    nombre_album CHAR(30),  
    discografia CHAR(30),  
    duracion INTEGER,  
    PRIMARY KEY (id_album),  
)
```

```
CREATE TABLE cds(  
    id_album INTEGER,  
    dvd_vivo CHAR(30),  
    PRIMARY KEY (id_album),  
    FOREIGN KEY (id_album) REFERENCES albums(id_album) ON DELETE CASCADE,  
)
```

```
CREATE TABLE cassettes(  
    id_album INTEGER,  
    id_cassettes INTEGER,  
    PRIMARY KEY (id_album),  
    FOREIGN KEY (id_album) REFERENCES albums(id_album) ON DELETE CASCADE,  
)
```

10.c

```
CREATE TABLE autores (  
    nombre CHAR(30),  
    direccion CHAR(30),  
    url CHAR(100),  
    PRIMARY KEY (nombre, dirección)  
)
```

```
CREATE TABLE producto (  
    id_producto INTEGER,  
    genero CHAR(30),  
    titulo CHAR(80),  
    precio REAL,  
    anio INTEGER,  
    PRIMARY KEY (id_producto)  
)
```

```
CREATE TABLE albums (  
    id_producto INTEGER,  
    discografica CHAR(40),  
    duracion INTEGER,  
    artista CHAR(40),  
    PRIMARY KEY (id_producto),  
    FOREIGN KEY (id_producto) REFERENCES producto(id_producto) ON DELETE CASCADE  
)
```

```
CREATE TABLE cassettes (  
    id_producto INTEGER,  
    PRIMARY KEY (id_producto),  
    FOREIGN KEY (id_producto) REFERENCES albums(id_producto) ON DELETE CASCADE  
)
```

```
CREATE TABLE cds (  
    id_producto INTEGER,  
    dvd_vivo CHAR(2),  
    PRIMARY KEY (id_producto),  
    FOREIGN KEY (id_producto) REFERENCES albums(id_producto) ON DELETE CASCADE  
)
```

```
CREATE TABLE libros (  
    id_producto INTEGER,  
    isbn CHAR(11),  
    nombre CHAR(30) NOT NULL,  
    PRIMARY KEY (id_producto),  
    FOREIGN KEY (id_producto) REFERENCES producto(id_producto) ON DELETE CASCADE,  
    FOREIGN KEY (nombre) REFERENCES editores(nombre)  
)
```

```
CREATE TABLE editores (  
    nombre CHAR(30),  
    direccion CHAR(30),
```

```
telefono INTEGER,  
url CHAR(80),  
PRIMARY KEY (nombre)  
)
```

```
CREATE TABLE almacenes (  
codigo INTEGER,  
telefono INTEGER,  
direccion CHAR(40),  
PRIMARY KEY (codigo)  
)
```

```
CREATE TABLE almacena (  
numero INTEGER,  
id_producto INTEGER,  
codigo INTEGER,  
PRIMARY KEY (id_producto, codigo),  
FOREIGN KEY (id_producto) REFERENCES producto(id_producto),  
FOREIGN KEY (codigo) REFERENCES almacenes(codigo)  
)
```

```
CREATE TABLE clientes (  
nombre CHAR(30),  
direccion CHAR(30),  
telefono INTEGER,  
direccion_email CHAR(50),
```

PRIMARY KEY (direccion_email)

)

CREATE TABLE **carro_compras** (

id_carro INTEGER,

direccion_email CHAR(30),

PRIMARY KEY (id_carro),

FOREIGN KEY (direccion_email) REFERENCES clientes(direccion_email)

)

CREATE TABLE **contiene** (

numero INTEGER,

id_carro INTEGER,

id_producto INTEGER,

PRIMARY KEY (id_carro, id_producto),

FOREIGN KEY (id_carro) REFERENCES carro_compras(id_carro),

FOREIGN KEY (id_producto) REFERENCES producto(id_producto)

)