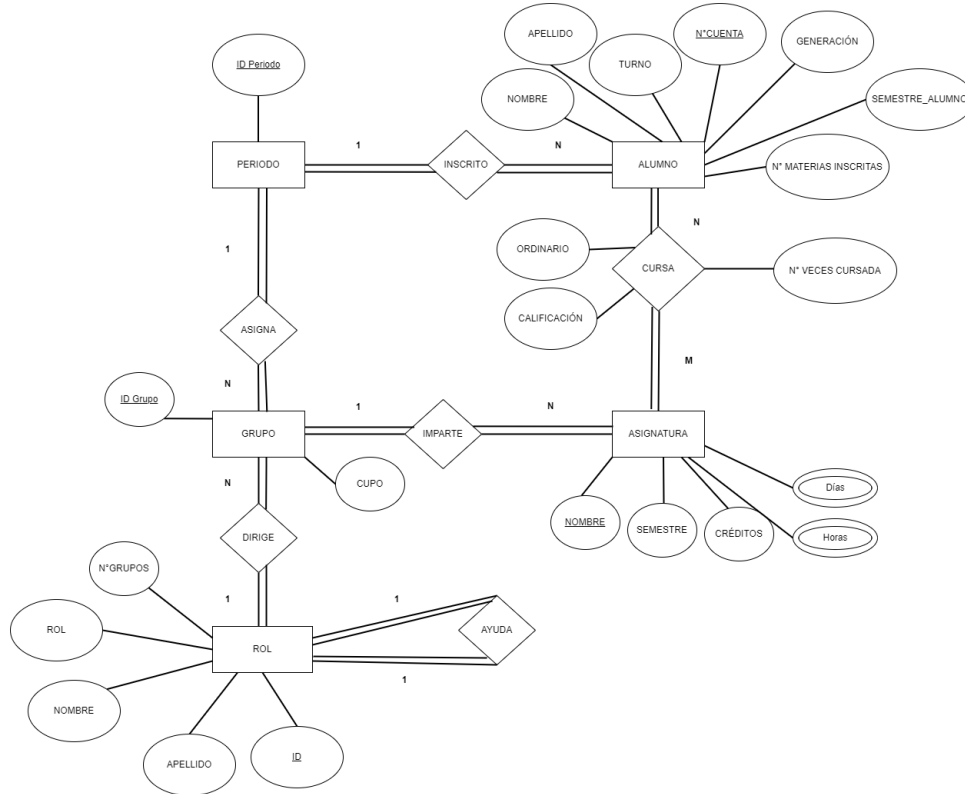
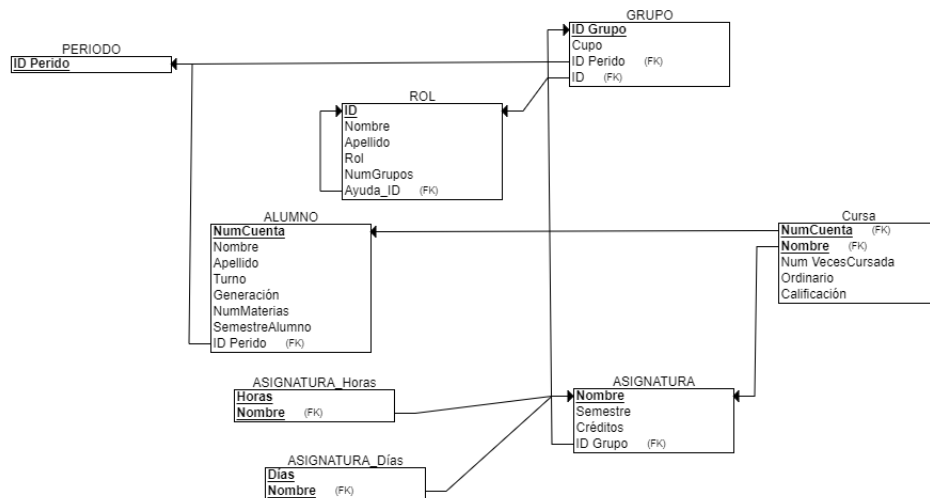


1. Modelo conceptual de la base de datos



2. Modelo relacional



3. Script completo y sin errores para la creación de todos los elementos que conforman el esquema de la base de datos.

(Versión inicial del script, posteriormente fue modificado agregando inserts y nuevas restricciones especificadas en los puntos 5 a 8 del reporte)

```
Script.sql x
home > valeria > Descargas > Proyectofdb > Script.sql
1 CREATE TABLE PERIODO
2 (
3     ID_Periodo INT NOT NULL,
4     CONSTRAINT pkPeriodo
5     PRIMARY KEY (ID_Periodo)
6 );
7
8 CREATE TABLE ALUMNO
9 (
10    Nombre VARCHAR(50) NOT NULL,
11    Apellido VARCHAR(50) NOT NULL,
12    NumCuenta INT NOT NULL,
13    Turno VARCHAR(10) NOT NULL,
14    Generacion INT NOT NULL,
15    NumMaterias INT NOT NULL,
16    SemestreAlumno CHAR(6) NOT NULL,
17    ID_Periodo INT NOT NULL,
18    CONSTRAINT pkAlumno
19    PRIMARY KEY (NumCuenta)
20 );
21
22 CREATE TABLE ROL
23 (
24    Nombre VARCHAR(50) NOT NULL,
25    Apellido VARCHAR(50) NOT NULL,
26    ID INT NOT NULL,
27    NumGrupos INT NOT NULL,
28    Rol BOOL NOT NULL,
29    ID_Ayuda INT NOT NULL,
30    CONSTRAINT pkRol
31    PRIMARY KEY (ID)
32 );
```

```
Scriptsql x
home > valeria > Descargas > Proyectofdb > Scripts.sql

33
34 CREATE TABLE GRUPO
35 (
36     ID_Grupo VARCHAR(5) NOT NULL,
37     Cupo INT NOT NULL,
38     ID_Periodo INT NOT NULL,
39     ID INT NOT NULL,
40     CONSTRAINT pkGrupo
41     PRIMARY KEY (ID_Grupo)
42 );
43
44 CREATE TABLE ASIGNATURA
45 (
46     Nombre VARCHAR(50) NOT NULL,
47     Creditos INT NOT NULL,
48     Semestre CHAR(6) NOT NULL,
49     ID_Grupo VARCHAR(5) NOT NULL,
50     CONSTRAINT pkAsignatura
51     PRIMARY KEY (Nombre)
52 );
53
54 CREATE TABLE Cursa
55 (
56     Num_VecesCursada INT NOT NULL,
57     Ordinatio BOOL NOT NULL,
58     NumCuenta INT NOT NULL,
59     Nombre VARCHAR(50) NOT NULL,
60     Calificacion INT NOT NULL,
61     CONSTRAINT pkCursa
62     PRIMARY KEY (NumCuenta, Nombre)
63 );
64
65 CONSTRAINT pkCursa
66 PRIMARY KEY (NumCuenta, Nombre)
67 );
68
69 CREATE TABLE ASIGNATURA_Dias
70 (
71     Dias VARCHAR(15) NOT NULL,
72     Nombre VARCHAR(50) NOT NULL,
73     CONSTRAINT pkAsigDias
74     PRIMARY KEY (Dias, Nombre)
75 );
76
77 CREATE TABLE ASIGNATURA_Horas
78 (
79     Horas VARCHAR(50) NOT NULL,
80     Nombre VARCHAR(50) NOT NULL,
81     CONSTRAINT pkAsigHoras
82     PRIMARY KEY (Horas, Nombre)
83 );
84
85 --Llaves Foráneas
86 ALTER TABLE Alumno add constraint fkAlumno FOREIGN KEY (ID_Periodo) REFERENCES PERIODO(ID_Periodo) ON UPDATE CASC
87 ALTER TABLE Asignatura add constraint fkAsignatura FOREIGN KEY (ID_Grupo) REFERENCES GRUPO(ID_Grupo) ON UPDATE CA
88 ALTER TABLE Cursa add constraint fkCursa1 FOREIGN KEY (NumCuenta) REFERENCES ALUMNO(NumCuenta) ON UPDATE CASCADE
89 ALTER TABLE Cursa add constraint fkCursa2 FOREIGN KEY (Nombre) REFERENCES ASIGNATURA(Nombre) ON UPDATE CASCADE ON
90 ALTER TABLE ASIGNATURA_Dias add constraint fkAsigDias FOREIGN KEY (Nombre) REFERENCES ASIGNATURA(Nombre) ON UPDAT
91 ALTER TABLE ASIGNATURA_Horas add constraint fkAsigHoras FOREIGN KEY (Nombre) REFERENCES ASIGNATURA(Nombre) ON UPD
92 ALTER TABLE Grupo add constraint fkGrupo1 FOREIGN KEY (ID_Periodo) REFERENCES PERIODO(ID_Periodo) ON UPDATE CASCA
93 ALTER TABLE Grupo add constraint fkGrupo2 FOREIGN KEY (ID) REFERENCES ROL(ID) ON UPDATE CASCADE ON DELETE CASCADE
94 ALTER TABLE Rol add constraint fkRol FOREIGN KEY (ID_Ayuda) REFERENCES ROL(ID) ON UPDATE CASCADE ON DELETE CASCA
95
96 int fkAlumno FOREIGN KEY (ID_Periodo) REFERENCES PERIODO(ID_Periodo) ON UPDATE CASCADE ON DELETE CASCADE;
97 straint fkAsignatura FOREIGN KEY (ID_Grupo) REFERENCES GRUPO(ID_Grupo) ON UPDATE CASCADE ON DELETE CASCADE;
98 nt fkCursa1 FOREIGN KEY (NumCuenta) REFERENCES ALUMNO(NumCuenta) ON UPDATE CASCADE ON DELETE CASCADE;
99 nt fkCursa2 FOREIGN KEY (Nombre) REFERENCES ASIGNATURA(Nombre) ON UPDATE CASCADE ON DELETE CASCADE;
100 d constraint fkAsigDias FOREIGN KEY (Nombre) REFERENCES ASIGNATURA(Nombre) ON UPDATE CASCADE ON DELETE CASCADE;
101 dd constraint fkAsigHoras FOREIGN KEY (Nombre) REFERENCES ASIGNATURA(Nombre) ON UPDATE CASCADE ON DELETE CASCADE;
102 nt fkGrupo1 FOREIGN KEY (ID_Periodo) REFERENCES PERIODO(ID_Periodo) ON UPDATE CASCADE ON DELETE CASCADE;
103 nt fkGrupo2 FOREIGN KEY (ID) REFERENCES ROL(ID) ON UPDATE CASCADE ON DELETE CASCADE;
104 fkRol FOREIGN KEY (ID_Ayuda) REFERENCES ROL(ID) ON UPDATE CASCADE ON DELETE CASCADE;
```

4. Script con Instrucciones Insert (Incluir sólo en los entregables)

5. Evidencia del funcionamiento de al menos 4 restricciones de integridad referencial

- (a) Tablas involucradas en la restricción.
- (b) FK de la tabla que referencia y PK de la tabla referenciada.
- (c) Justificación del trigger de integridad referencial elegido.
- (d) Instrucción UPDATE o DELETE que permita evidenciar que la restricción está funcionando.
- (e) Captura de pantalla con el resultado de la instrucción que muestre que la restricción está funcionando.

Restricción 1

- (a) *Grupo y Periodo*
- (b) *FK de Grupo: fkGrupo1 y PK de Periodo: pkPeriodo*
- (c) Usando ON UPDATE CASCADE actualizamos PK en Periodo, lo que a hacer que se actualicen en cascada los datos de FK en Grupo
- (d) UPDATE Periodo SET ID Periodo = '20232' WHERE ID Periodo = '20231';

(e)

```
sistenaufbd-> SELECT * FROM PERIODO;
id_periodo
-----
20222
20221
20212
20211
20202
20201
20192
20191
20182
20181
20172
20171
20162
20161
20152
20151
20142
20141
20132
20131
20232
(21 rows)
```

id_grupo	cupo	id_periodo	id
2100M	3	20222	6
2200V	3	20222	7
2300V	3	20222	8
2400M	3	20222	9
2500M	3	20222	10
3100V	2	20221	11
3200V	2	20221	12
3300M	2	20221	13
3400M	2	20221	14
3500V	2	20221	15
4100V	2	20212	16
4200M	2	20212	17
4300M	2	20212	18
4400V	2	20212	19
4500V	2	20212	20
5100M	2	20211	1
5200M	2	20211	2
5300V	2	20211	3
5400V	2	20211	4
5500M	2	20211	5
6100M	2	20202	6
6200V	2	20202	7
6300V	2	20202	8
7100M	2	20201	9
7200M	2	20201	10
7300V	2	20201	11
8100V	2	20192	12
8200M	2	20192	13
6661M	2	20202	14
6662V	2	20202	15
7771V	2	20201	16
7772M	2	20201	17
8881M	2	20192	18
8882V	2	20192	19
8883V	2	20192	20
3100V	2	20221	11
3200V	2	20221	12
3300M	2	20221	13
3400M	2	20221	14
3500V	2	20221	15
4100V	2	20212	16
4200M	2	20212	17
4300M	2	20212	18
4400V	2	20212	19
4500V	2	20212	20
5100M	2	20211	1
5200M	2	20211	2
5300V	2	20211	3
5400V	2	20211	4
5500M	2	20211	5
6100M	2	20202	6
6200V	2	20202	7
6300V	2	20202	8
7100M	2	20201	9
7200M	2	20201	10
7300V	2	20201	11
8100V	2	20192	12
8200M	2	20192	13
6661M	2	20202	14
6662V	2	20202	15
7771V	2	20201	16
7772M	2	20201	17
8881M	2	20192	18
8882V	2	20192	19
8883V	2	20192	20
1100M	3	20232	1
1200M	3	20232	2
1300V	3	20232	3
1400V	3	20232	4
1500M	3	20232	5

(40 rows)

Restricción 2

- Alumno* y *Periodo*
- FK de Alumno*: fkAlumno y *PK de Periodo*: pkPeriodo
- Usando ON DELETE DEFAULT la PK en Periodo va a pasar a su valor establecido como default, lo que a hacer que también ocurra en la FK en Alumno. La PK ID Periodo tiene como default '20232'.
- DELETE FROM Periodo WHERE ID Periodo = '20202';
- La tabla Periodo:

id_periodo
20222
20221
20212
20211
20201
20192
20191
20182
20181
20172
20171
20162
20161
20152
20151
20142
20141
20132
20131
20231
20232

La tabla Alumno:

Angel	Cuevas	320000022	Vespertino	2019	4	1	20231
Aarón	Álvarez	320000023	Vespertino	2019	4	1	20231
Abigail	Solano	320000024	Vespertino	2019	4	1	20231
Abisay	Paredes	320000025	Vespertino	2019	4	1	20231
Abraham	Mendoza	320000026	Vespertino	2019	4	1	20231
Axel	Tellez	320000027	Vespertino	2019	4	1	20231
Adair	Ibarra	320000028	Vespertino	2019	4	1	20231
Adan	Ramos	320000029	Vespertino	2019	4	1	20231
Adolfo	Zamora	320000030	Vespertino	2019	4	1	20231
Karen	Velez	317000031	Vespertino	2019	4	6	20232
Victor	Cuapio	317000032	Vespertino	2019	4	6	20232

Restricción 3

- Rol*
- FK de Rol*: fkRol y *PK de Rol*: pkRol
- Usando ON UPDATE CASCADE actualizamos PK en Rol, lo que a hacer que se actualicen en cascada los datos de FK en Rol. El que sólo ocupemos 1 tabla en este caso es porque Rol es una asociación recursiva en el diagrama ER, haremos esta actualización para ver si los datos sí se actualizan en estos casos.
- UPDATE Rol SET ID = '300' WHERE ID = '1';

nombre	apellido	id	numgrupos	rol	id_ayuda
Aurora	Rodriguez	2	2	t	22
Beatriz	Manthroppe	3	2	t	23
Elfrida	Mull	4	2	t	24
Anette	Finlan	5	2	t	25
Carmelita	Perez	6	2	t	26
Evin	Cromett	7	2	t	27
Brittni	Verden	8	2	t	28
Gracia	Filton	9	2	t	29
Irwin	Orme	10	2	t	30
Clarance	Danshin	11	2	t	31
Andriette	Turtle	12	2	t	32
Adan	Melin	13	2	t	33
Felicia	Buckleigh	14	2	t	34
Freddy	Nesby	15	2	t	35
Lucilla	Rossbrook	16	2	t	36
Bianca	Swift	17	2	t	37
Monica	Canwithen	18	2	t	38
Stuart	Alwyn	19	2	t	39
Juan	Alvarez	20	2	t	40
Ernesto	Marquez	300	2	t	21
Cristian	Mendez	21	2	f	300
Axel	Blower	22	2	f	2
Ingrid	Arnowitz	23	2	f	3
Martha	Catling	24	2	f	4
Madelaine	Oldnall	25	2	f	5
Paulette	Hambers	26	2	f	6
Jose	Lucio	27	2	f	7
Gael	McMenamie	28	2	f	8
Lucine	Dast	29	2	f	9
Emily	Gilbert	30	2	f	10
Selene	Cabello	31	2	f	11
Lidia	Talavera	32	2	f	12
Socorro	Rodriguez	33	2	f	13
Raul	Pineda	34	2	f	14
Angel	Gomez	35	2	f	15

(e)

Restricción 4

- Asignatura y Asignatura Dias*
- FK de Asignatura Dias: fkAsigDias y PK de Asignatura: pkAsignatura*
- Usando ON UPDATE CASCADE actualizamos PK en Asignatura, lo que a hacer que se actualicen en cascada los datos de FK en Asignatura Dias. La relación Asignatura Dias es el resultado de tener en el diagrama ER un atributo multivaluado para Asignatura, haremos esta actualización para ver si los datos sí se actualizan en estos casos.
- UPDATE Asignatura SET nombre = 'Algebra Superior III' WHERE nombre = 'Algebra Superior I';

Estructuras de Datos	12	2	2200V
Graficas y Juegos	10	2	2300V
Inglés II	4	2	2400M
Matemáticas para las Ciencias Aplicadas II	12	2	2500M
Algebra Lineal I	10	3	3100V
Inglés III	4	3	3200V
Matemáticas para las Ciencias Aplicadas III	12	3	3300M
Modelado y Programación	10	3	3400M
Probabilidad I	10	3	3500V
Autómatas y Lenguajes Formales	10	4	4100V
Inglés IV	4	4	4200M
Lógica Computacional	10	4	4300M
Matemáticas para las Ciencias Aplicadas IV	12	4	4400V
Organización y Arquitectura de Computadoras	10	4	4500V
Análisis de Algoritmos	10	5	5100M
Lenguajes de Programación	10	5	5200M
Computación Distribuida	10	5	5300V
Fundamentos de Bases de Datos	10	5	5400V
Inglés V	4	5	5500M
Ingeniería de Software	10	6	6100M
Inteligencia Artificial	10	6	6200V
Sistemas Operativos	10	6	6300V
Compiladores	10	7	7100M
Complejidad Computacional	10	7	7200M
Redes de Computadoras	10	7	7300V
Computación Concurrente	10	8	8100V
Criptografía y Seguridad	10	8	8200M
Algoritmos Paralelos	10	6	6661M
Análisis Numérico	10	6	6662V
Computación Cuántica I	10	7	7771V
Cómputo Evolutivo	10	7	7772M
Estadística I	10	8	8881M
Geometría Moderna I	10	8	8882V
Lenguajes de Programación II	10	8	8883V
Algebra Superior III	10	1	1100M
(40 rows)			
Viernes	Fundamentos de Bases de Datos		
Lunes	Inglés V		
Miercoles	Inglés V		
Martes	Ingeniería de Software		
Jueves	Ingeniería de Software		
Miercoles	Inteligencia Artificial		
Viernes	Inteligencia Artificial		
Lunes	Sistemas Operativos		
Miercoles	Sistemas Operativos		
Martes	Compiladores		
Jueves	Compiladores		
Miercoles	Complejidad Computacional		
Viernes	Complejidad Computacional		
Lunes	Redes de Computadoras		
Miercoles	Redes de Computadoras		
Martes	Computación Concurrente		
Jueves	Computación Concurrente		
Miercoles	Criptografía y Seguridad		
Viernes	Criptografía y Seguridad		
Lunes	Algoritmos Paralelos		
Miercoles	Algoritmos Paralelos		
Martes	Análisis Numérico		
Jueves	Análisis Numérico		
Miercoles	Computación Cuántica I		
Viernes	Computación Cuántica I		
Lunes	Cómputo Evolutivo		
Miercoles	Cómputo Evolutivo		
Martes	Estadística I		
Jueves	Estadística I		
Miercoles	Geometría Moderna I		
Viernes	Geometría Moderna I		
Lunes	Lenguajes de Programación II		
Miercoles	Lenguajes de Programación II		
Lunes	Algebra Superior III		
Miercoles	Algebra Superior III		
(80 rows)			

6. Evidencia del funcionamiento de al menos 6 restricciones para “atributos” de varias tablas.

- (a) Tabla elegida
- (b) Atributo elegido
- (c) Breve descripción de la restricción
- (d) Instrucción para la creación de la restricción.
- (e) Instrucción que permita evidenciar que la restricción esta funcionando.
- (f) Captura de pantalla con el resultado de la instrucción que muestre que la restricción está funcionando.
- (g) (No cuentan las restricciones NOT NULL)

Restricción 1

- (a) *Alumno*
- (b) *Generacion*
- (c) Restricción para que la generación sea mayor a 2013
- (d) ALTER TABLE Alumno ADD CONSTRAINT Generacion CHECK (Generacion > 2013);
- (e) UPDATE Alumno SET generacion = '2012' WHERE semestrealumno = '1';

- (f)

Restricción 2

- (a) *Asignatura*
- (b) *Creditos*
- (c) Restricción para que los créditos se encuentren en un rango de: 4,5,6,7,8,9,10,11,12
- (d) ALTER TABLE Asignatura ADD CONSTRAINT Creditos CHECK (Creditos > 3 AND Creditos < 13);
- (e) UPDATE Asignatura SET creditos = '20' WHERE nombre = 'Algebra Superior I';

(f)

```
sistematfbd=> UPDATE Asignatura SET credits = '20' WHERE nombre = 'Algebra Superior I';  
ERROR: new row for relation "asignatura" violates check constraint "credits"  
DETALLE: Failing row contains (Algebra Superior I, 20, 1, 1100M).  
sistematfbd=>
```

Restricción 3

- (a) *Alumno*
- (b) *Turno*
- (c) Restricción para que los turnos sean sólo Matutino o Vespertino
- (d) ALTER TABLE Alumno ADD CONSTRAINT Turno CHECK(Turno = 'Matutino' OR Turno = 'Vespertino');
- (e) UPDATE Alumno SET turno = 'Mixto' WHERE turno = 'Vespertino';

(f)

```
sistematfbd=> UPDATE Alumno SET turno = 'Mixto' WHERE turno = 'Vespertino';  
ERROR: new row for relation "alumno" violates check constraint "turno"  
DETALLE: Failing row contains (Yusneli, Ulloa, 320000021, Mixto, 2019, 4, 1, 20231).  
sistematfbd=>
```

Restricción 4

- (a) *Rol*
- (b) *NumGrupos*
- (c) Restricción para que los grupos para cada rol no pasen de 3
- (d) ALTER TABLE Rol ADD CONSTRAINT NumGrupos CHECK (NumGrupos <= 3);
- (e) UPDATE Rol SET NumGrupos = '7' WHERE NumGrupos = '2';

(f)

```
sistematfbd=> UPDATE Rol SET NumGrupos = '4' WHERE NumGrupos = '2';  
ERROR: new row for relation "rol" violates check constraint "numgrupos"  
DETALLE: Failing row contains (Ernesto, Marquez, 1, 4, t, 21).  
sistematfbd=>
```

Restricción 5

- (a) *Grupo*
- (b) *Cupo*
- (c) Restricción para que los cupos grupos para cada grupo no pasen de 10
- (d) ALTER TABLE Grupo ADD CONSTRAINT Cupo CHECK (Cupo <= 10);
- (e) UPDATE Grupo SET Cupo = '11' WHERE Cupo = '3';

(f)

```
sistematfbd=> UPDATE Grupo SET Cupo = '11' WHERE Cupo = '3';  
ERROR: new row for relation "grupo" violates check constraint "cupo"  
DETALLE: Failing row contains (1100M, 11, 20231, 1).  
sistematfbd=>
```

Restricción 6

- (a) *Alumno*
- (b) *NumMaterias*
- (c) Restricción para que el número de materias por alumno no pase de 6
- (d) ALTER TABLE Alumno ADD CONSTRAINT NumMaterias CHECK (NumMaterias < 7);
- (e) UPDATE Alumno SET NumMaterias = '7' WHERE NumMaterias = '4';

(f)

```
sistemafbd=> UPDATE Alumno SET NumMaterias = '7' WHERE NumMaterias = '4';  
ERROR: new row for relation "alumno" violates check constraint "nummaterias"  
DETALLE: Falling row contains (Sebastián, Torres, 320000001, Matutino, 2019, 7, 1, 20231).  
sistemafbd=>
```

7. Evidencia de la creación de al menos tres dominios personalizados.

- (a) Tabla elegida
- (b) Atributo elegido
- (c) Breve descripción del dominio
- (d) Instrucción para la creación del dominio personalizado
- (e) Captura de pantalla de la estructura de la tabla donde se muestre el dominio personalizado en uso.

Dominio 1

- (a) *Alumno* y *Asignatura*
- (b) De *Alumno* el atributo es *SemestreAlumno*, mientras que de *Asignatura* es *Semestre*
- (c) Creación de un tipo *SemestreType* en base a otro tipo que es CHAR(6)
- (d) CREATE DOMAIN *SemestreType* AS CHAR(6);

(e)

```
sistematfbd=> \d asignatura
Table "public.asignatura"
Column | Type | Collation | Nullable | Default
-----|-----|-----|-----|-----
nombre | character varying(50) | | not null |
creditos | integer | | not null |
semestre | semestertype | | not null |
id_grupo | character varying(5) | | not null |
Indexes:
    "pkasignatura" PRIMARY KEY, btree (nombre)
Foreign-key constraints:
    "fkasignatura" FOREIGN KEY (id_grupo) REFERENCES grupo(id_grupo) ON UPDATE CASCADE ON DELETE CASCADE
Referenced by:
    TABLE "asignatura_dias" CONSTRAINT "fkasigdias" FOREIGN KEY (nombre) REFERENCES asignatura(nombre) ON UPDATE CASCADE ON DELETE CASCADE
    TABLE "asignatura_horas" CONSTRAINT "fkasighoras" FOREIGN KEY (nombre) REFERENCES asignatura(nombre) ON UPDATE CASCADE ON DELETE CASCADE
    TABLE "cursa" CONSTRAINT "fkursa2" FOREIGN KEY (nombre) REFERENCES asignatura(nombre) ON UPDATE CASCADE ON DELETE CASCADE

sistematfbd=> \d alumno
Table "public.alumno"
Column | Type | Collation | Nullable | Default
-----|-----|-----|-----|-----
nombre | character varying(50) | | not null |
apellido | character varying(50) | | not null |
numcuenta | integer | | not null |
turno | character varying(10) | | not null |
generacion | integer | | not null |
nummaterias | integer | | not null |
semestrealumno | semestertype | | not null |
id_periodo | integer | | not null |
Indexes:
    "pkalumno" PRIMARY KEY, btree (numcuenta)
Foreign-key constraints:
    "fkalumno" FOREIGN KEY (id_periodo) REFERENCES periodo(id_periodo) ON UPDATE CASCADE ON DELETE CASCADE
Referenced by:
    TABLE "cursa" CONSTRAINT "fkursa1" FOREIGN KEY (numcuenta) REFERENCES alumno(numcuenta) ON UPDATE CASCADE ON DELETE CASCADE
```

Dominio 2

- Alumno*
- De *Alumno* el atributo es *Turno*
- Creación de un tipo TurnoType en base a otro tipo que es VARCHAR(10)
- CREATE DOMAIN TurnoType AS VARCHAR(10);

(e)

```
sistematfbd=> CREATE DOMAIN TurnoType AS VARCHAR(10);
CREATE DOMAIN

sistematfbd=> \d alumno
Table "public.alumno"
Column | Type | Collation | Nullable | Default
-----|-----|-----|-----|-----
nombre | character varying(50) | | not null |
apellido | character varying(50) | | not null |
numcuenta | integer | | not null |
turno | turnotype | | not null |
generacion | integer | | not null |
nummaterias | integer | | not null |
semestrealumno | semestertype | | not null |
id_periodo | integer | | not null |
Indexes:
    "pkalumno" PRIMARY KEY, btree (numcuenta)
Foreign-key constraints:
    "fkalumno" FOREIGN KEY (id_periodo) REFERENCES periodo(id_periodo) ON UPDATE CASCADE ON DELETE CASCADE
Referenced by:
    TABLE "cursa" CONSTRAINT "fkursa1" FOREIGN KEY (numcuenta) REFERENCES alumno(numcuenta) ON UPDATE CASCADE ON DELETE CASCADE
```

Dominio 3

- Grupo*
- De *Grupo* el atributo es *Id Grupo*
- Creación de un tipo GrupoType en base a otro tipo que es VARCHAR(5)
- CREATE DOMAIN GrupoType AS VARCHAR(5);

(e)

```
sistemafbd=> CREATE DOMAIN GrupoType AS VARCHAR(5);
CREATE DOMAIN

Table "public.grupo"
Column | Type | Collation | Nullable | Default
-----|-----|-----|-----|-----
id_grupo | grupotype | | not null | 
cupo | integer | | not null | 
id_periodo | integer | | not null | 
id | integer | | not null | 
Indexes:
    "pkgrupo" PRIMARY KEY, btree (id_grupo)
Foreign-key constraints:
    "fkgrupo1" FOREIGN KEY (id_periodo) REFERENCES periodo(id_periodo) ON UPDATE CASCADE ON DELETE CASCADE
    "fkgrupo2" FOREIGN KEY (id) REFERENCES rol(id) ON UPDATE CASCADE ON DELETE CASCADE
Referenced by:
    TABLE "asignatura" CONSTRAINT "fkasignatura" FOREIGN KEY (id_grupo) REFERENCES grupo(id_grupo) ON UPDATE CASCADE ON DELETE CASCADE
```

8. Evidencia del funcionamiento de al menos 2 restricciones para “tuplas” en diferentes tablas.

- (a) Tabla elegida
- (b) Breve descripción de la restricción
- (c) Instrucción para la creación de la restricción
- (d) Instrucción que permita evidenciar que la restricción esta funcionando.
- (e) Captura de pantalla con el resultado de la instrucción que muestre que la restricción está funcionando.

Restricción 1

- (a) *Alumno*
- (b) Restricción para que la generación sea menor que el periodo en el que se encuentra el alumno.
- (c) ALTER TABLE Alumno ADD CONSTRAINT ID Periodo CHECK (Generacion < ID Periodo);
- (d) UPDATE Alumno SET ID Periodo = '2018' WHERE Generacion = '2019';

(e)

```
sistemafbd=> UPDATE Alumno SET ID_Periodo = '2018' WHERE Generacion = '2019';
ERROR: new row for relation "alumno" violates check constraint "id_periodo"
DETALLE: Failing row contains (Adrian, Villeda, 319000001, Matutino, 2019, 4, 3, 2018).
sistemafbd=>
```

Restricción 2

- (a) *Rol*
- (b) Restricción para que el ID del profesor (ID) no sea el mismo que el ID de su ayudante (ID Ayuda).

(c) ALTER TABLE Rol ADD CONSTRAINT ID CHECK (ID <> ID Ayuda);

(d) UPDATE Rol SET ID = '1' WHERE ID AYUDA = '1';

(e)

```
sistemafbd=> UPDATE Rol SET ID = '1' WHERE ID_AYUDA = '1';  
ERROR: new row for relation "rol" violates check constraint "id"  
DETALLE: Failing row contains (Cristian, Mendez, 1, 2, f, 1).  
sistemafbd=> █
```