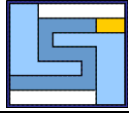


D/Dª <<Jose Maria, Villamor Carrasco>> con NIF <<30259150M>> y alumno/a de la asignatura IISSI-1 perteneciente al grupo <<GRUPO 1>>.

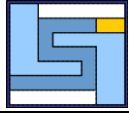
Asegura que es el/la autor/a de este trabajo enviado a través de la plataforma de enseñanza virtual, y que ha realizado dicho trabajo personalmente y no es una copia de ningún otro trabajo. Asimismo, asegura que no ha entregado o mostrado su trabajo a terceras personas.

Contenido

1. Modelo Relacional.....	3
2. Modelo Tecnológico.	5
2.1 Requisitos de información.....	5
2.1.1. Tabla Degrees.....	5
2.1.2. Tabla X.....	5
2.2 Requisitos funcionales.	14
2.2.1 RF-001. Añadir nota	14
2.2.2 RF-002. Citas de una tutoría	14
2.2.3 RF-003. Asignación de asignaturas	14
2.2.4 RF-005 Profesor que imparte más créditos en una asignatura	14
2.2.5 RF-006. Borrar notas de un alumno	14
2.2.6 RF-011. Nota media de alumno	15
2.2.7 RF-036 Listado de alumnos de una asignatura por grupos.....	15
2.2.8 RF-037 Listado de grupos por profesor y asignatura	16
2.2.9 RF-038 Listado de despachos con más de un profesor	16
2.2.10 RF-039 Listado de tutorías pendientes de cada profesor.....	16
2.3 Reglas de negocio.	17
2.3.1 RN-001 Limitación créditos docentes.....	17
2.3.2 RN-006 Matrícula de honor	17
2.3.3 RN-007 Nota por alumno	18
2.3.4 RN-010 Cambios bruscos en notas.....	18
2.3.5 RN-019 Alumno pertenece a grupo.....	19
2.3.6 RN-020 Profesor de teoría y prácticas	19
2.3.7 RN-021 Aulas de teoría y prácticas.....	19
2.3.8 RN-022 Citas para tutorías.....	19
3. Anexo con todo el código SQL.....	21



--



1. Modelo Relacional

Degrees(!degreeId, #name, #years)

Departments(!departmentId, #name)

Subjects(!subjectId, @degreeId, @departmentId, #name, #acronym, #credits,
#course, #type,)

Spaces(!spaceId, #name, #floor, #capacity) Offices(!officeId, @spaceId,
@isShared, @isThereFreeSpace) Classrooms(!classroom, @spaceId,
#loudspeakers, #projector)

Groups(!groupId, @subjectId, @classroomId, #name, activity, year)

Students(!studentId, !accessMethod, #DNI, firstname, lastName, birthDate,
email)

Grades(!gradeId, @studentId, @groupId, #value, gradecall, withhonour)

Teachers(!teacherId, @officeId, @departmentId, #DNI, firstName, lastName,
birthDate, email, category)

Tutorials(!tutorialId, @teacherId, #dayweek, startTime, endTime)

CORREGIR EL CAMBIO DE IDIOMA

Grados(!gradold, #nombre, años, ncredits)

Departamentos(!departamentold, #nombre, numeroAsignatura)

Tutorias(!tutorialId, @profesorId, diadelaSemana, hora comienzo, hora fin)

Profesores(!profesorId, @despachold, @departamentold, , dni, nombre,
apellido, fechaNacimiento, email categoria)

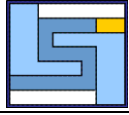
Alumnos(!alumnosId, #dni, nombre, apellidos, metodoacceso, , fechaNacimiento,
email)

Aulas(!AulaId, #nombre, megafonia, proyector, teoria, laboratorio)

Notas (!notald, @alumnold, @grupold, puntuación)

Grupos(!Grupold, @asignaturald, nombre, añoAcademico, credits)

Espacios(!despachold, #nombre planta, capacidad)



Introducción a la Ingeniería del Software y
los Sistemas de Información I
Curso 2020-21. Entregable Modelado Tecnológico

Cita(!citald,@horatutoriald,@profesorId,@alumnosId,hora,fecha)

Imparte(!ImpartId,creditos,#(@profesorId,@grupold))

AsignaturaGrado(!asignaturaGradold,#(@asignaturald,@gradold))

GrupoAsignatura(!grupoAsignaturald,#(@grupold,@asignaturald))

AlumnoNota(!alumnoNota,#(@alumnold,@notald))

NotaGrupo(!notaGrupold,#(@notald,@grupold))

AlumnoCita(!alumnoCitald,#(@alumnold,@citald))

CitaTutoria(!citaTutoriald,#(@citald,@tutoriald))

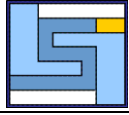
ProfesorTutoria(!profesorTutoria,#(@profesorId,@tutoriald))

DespachoProfesor(!despachoProfesorId,#(@despachold,@profesorId))

ProfesorDepartamento(!profesorDepartamentold,#(@profesorId,@departamentold))

DepartamentoAsignatura(!departamentoAsignaturald,#(@departamentold,@Asignaturald))

--



2. Modelo Tecnológico.

2.1 Requisitos de información

2.1.1. Tabla Degrees

```
DROP TABLE IF EXISTS DEGREES;
/* Tabla de Grados */
CREATE TABLE Degrees (
    degreeId INT AUTO_INCREMENT,
    name VARCHAR(60),
    years INT DEFAULT (4),
    PRIMARY KEY(degreeId)
);
/* Datos de Prueba */
INSERT INTO DEGREES(name, years) VALUES ('Grado en Ingeniería
Informática - Tecnologías Informáticas', 4);
INSERT INTO DEGREES(name, years) VALUES ('Grado en Ingeniería
Informática - Ingeniería del Software', 4);
INSERT INTO DEGREES(name, years) VALUES ('Grado en Ingeniería
Informática - Ingeniería de Computadores', 4);
```

test.degrees: 3 filas en total (aproximadamente)

degreeId	name	years
1	Grado en Ingeniería Informática - Tecnologías Informáticas	4
2	Grado en Ingeniería Informática - Ingeniería del Software	4
3	Grado en Ingeniería Informática - Ingeniería de Computadores	4

Tabla 1. Degrees

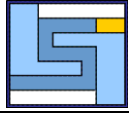
2.1.2. Tabla X

```
SET FOREIGN_KEY_CHECKS=0;
DROP TABLE IF EXISTS Grados;
DROP TABLE IF EXISTS Asignaturas;
DROP TABLE IF EXISTS Grupos;
DROP TABLE IF EXISTS Alumnos;
DROP TABLE IF EXISTS GruposAlumnos;
DROP TABLE IF EXISTS Profesores;
DROP TABLE IF EXISTS Departamentos;
DROP TABLE IF EXISTS Despachos;
DROP TABLE IF EXISTS Aulas;
DROP TABLE IF EXISTS Tutorias;
DROP TABLE IF EXISTS Notas;

DROP TABLE IF EXISTS Citas ;
DROP TABLE IF EXISTS Imparte ;
DROP TABLE IF EXISTS AlumnoCita ;
DROP TABLE IF EXISTS CitaTutoria ;
DROP TABLE IF EXISTS ProfesorTutoria ;
DROP TABLE IF EXISTS DespachoProfesor ;

DROP TABLE IF EXISTS ProfesorDepartamento ;
DROP TABLE IF EXISTS DepartamentoAsignatura ;

DROP TABLE IF EXISTS GrupoAsignatura ;
DROP TABLE IF EXISTS AsignaturaGrado ;
DROP TABLE IF EXISTS AlumnoNota ;
```



```
DROP TABLE IF EXISTS NotaGrupo ;
```

```
SET FOREIGN_KEY_CHECKS=1;
```

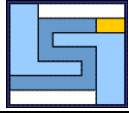
```
CREATE TABLE Grados (  
gradoId INT NOT NULL AUTO_INCREMENT,  
nombre VARCHAR(60) NOT NULL UNIQUE,  
anyos INT DEFAULT(4) NOT NULL,  
nCreditos INT NOT NULL,  
PRIMARY KEY (gradoId),  
  
);
```

```
CREATE TABLE Asignaturas (  
asignaturaId INT NOT NULL AUTO_INCREMENT,  
nombre VARCHAR(100) NOT NULL UNIQUE,  
acronimo VARCHAR(8) NOT NULL UNIQUE,  
creditos INT NOT NULL,  
curso INT NOT NULL,  
tipo VARCHAR(20) NOT NULL,  
gradoId INT NOT NULL,  
PRIMARY KEY (asignaturaId),  
FOREIGN KEY (gradoId) REFERENCES Grados (gradoId),  
CONSTRAINT negativeSubjectcreditos CHECK (creditos > 0),  
CONSTRAINT invalidSubjectcurso CHECK (curso > 0 AND curso < 6),  
CONSTRAINT invalidSubjecttipo CHECK (tipo IN ('Formacion Basica',  
'Optativa',  
'Obligatoria'))  
);
```

```
CREATE TABLE Despachos (  
despachoId INT NOT NULL AUTO_INCREMENT,  
nombre VARCHAR(100) NOT NULL UNIQUE,  
Planta INT NOT NULL UNIQUE,  
PRIMARY KEY (despachoId)  
);
```

```
CREATE TABLE Aulas (  
aulaId INT NOT NULL AUTO_INCREMENT,  
nombre VARCHAR(100) NOT NULL UNIQUE,  
planta INT NOT NULL UNIQUE,  
megafonia BOOLEAN NOT NULL,  
proyector BOOLEAN NOT NULL,  
teoria BOOLEAN NOT NULL,  
laboratorio BOOLEAN NOT NULL,  
PRIMARY KEY (aulaId)  
);
```

```
CREATE TABLE Grupos (  
grupoId INT NOT NULL AUTO_INCREMENT,  
asignaturaId INT NOT NULL,  
nombre VARCHAR(30) NOT NULL,  
anyos INT NOT NULL,  
creditos INT NOT NULL,  
  
PRIMARY KEY (grupoId),  
FOREIGN KEY (asignaturaId) REFERENCES Asignaturas (asignaturaId),
```



```
UNIQUE (nombre, anyos, asignaturaId),
CONSTRAINT negativeGroupanyos CHECK (anyos > 0),
CONSTRAINT invalidGroupactividad CHECK (actividad IN ('Teoria',
'Laboratorio'))
);

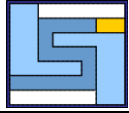
CREATE TABLE Alumnos (
alumnoId INT NOT NULL AUTO_INCREMENT,
dni CHAR(9) NOT NULL UNIQUE,
nombre VARCHAR(100) NOT NULL,
apellidos VARCHAR(100) NOT NULL,
metodoAcceso VARCHAR(30) NOT NULL,
birthDate DATE NOT NULL,
email VARCHAR(250) NOT NULL ,
PRIMARY KEY (alumnoId),
CONSTRAINT invalidmetodoAcceso CHECK (metodoAcceso IN ('Selectividad',
'Ciclo',
'Mayor',
'Titulado Extranjero'))
);

CREATE TABLE GruposAlumnos (
grupoalumnoId INT NOT NULL AUTO_INCREMENT,
grupoId INT NOT NULL,
alumnoId INT NOT NULL,
PRIMARY KEY (grupoalumnoId),
FOREIGN KEY (grupoId) REFERENCES Grupos (grupoId) ON DELETE CASCADE,
FOREIGN KEY (alumnoId) REFERENCES Alumnos (alumnoId),
UNIQUE (grupoId, alumnoId)
);

CREATE TABLE Notas (
notaId INT NOT NULL AUTO_INCREMENT,
puntuacion DOUBLE NOT NULL,
alumnoId INT NOT NULL,
grupoId INT NOT NULL,
PRIMARY KEY (notaId),
FOREIGN KEY (alumnoId) REFERENCES Alumnos (alumnoId),
FOREIGN KEY (grupoId) REFERENCES Grupos (grupoId) ON DELETE CASCADE
);

CREATE TABLE Departamentos (
departamentoId INT NOT NULL AUTO_INCREMENT,
nombre VARCHAR(100) NOT NULL UNIQUE,
numeroAsiganturas INT NOT NULL,
PRIMARY KEY (departamentoId)
);

CREATE TABLE Profesores (
profesorId INT NOT NULL AUTO_INCREMENT,
despachoId INT NOT NULL,
departamentoId INT NOT NULL,
dni CHAR(9) NOT NULL UNIQUE,
nombre VARCHAR(100) NOT NULL ,
apellidos VARCHAR(100) NOT NULL ,
fechaNacimiento DATE NOT NULL ,
emailProfesor VARCHAR(100) NOT NULL,
categoria VARCHAR(100) NOT NULL,
```



```
PRIMARY KEY (profesorId),  
FOREIGN KEY (despachoId) REFERENCES Despachos (despachoId),  
FOREIGN KEY (departamentoId) REFERENCES Departamentos  
(departamentoId),  
CONSTRAINT categoriaInvalida CHECK (categoria IN ('Catedratico',  
'Titular de Universidad',  
'Profesor Contrato Doctor',  
'Profesor Ayudante Doctor'))  
);
```

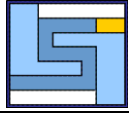
```
CREATE TABLE Tutorias(  
tutoriaId INT NOT NULL AUTO_INCREMENT,  
profesorId INT NOT NULL,  
dia DATE NOT NULL,  
horaComienzo TIME NOT NULL,  
horaFin TIME NOT NULL,  
PRIMARY KEY(tutoriaId),  
FOREIGN KEY (profesorId) REFERENCES Profesores(profesorId)  
);
```

```
CREATE TABLE Citas(  
citaId INT NOT NULL AUTO_INCREMENT,  
tutoriaId INT NOT NULL,  
alumnoId INT NOT NULL,  
hora TIME NOT NULL,  
fecha DATE NOT NULL,  
PRIMARY KEY(citaId),  
FOREIGN KEY (tutoriaId) REFERENCES Tutorias(tutoriaId),  
FOREIGN KEY (alumnoId) REFERENCES Alumnos(alumnoId)  
);
```

```
CREATE TABLE Imparte(  
ImparteId INT NOT NULL AUTO_INCREMENT ,  
creditos DOUBLE NOT NULL,  
profesorId INT NOT NULL,  
grupoid INT NOT NULL,  
PRIMARY KEY (ImparteId),  
FOREIGN KEY (profesorId) REFERENCES Profesores(profesorId),  
FOREIGN KEY (grupoid) REFERENCES Grupos(grupoid)  
  
);
```

```
CREATE TABLE AlumnoCita(  
alumnoCitaId INT NOT NULL AUTO_INCREMENT,  
alumnoId INT NOT NULL,  
citaId INT NOT NULL,  
PRIMARY KEY (alumnoCitaId),  
FOREIGN KEY (alumnoId) REFERENCES Alumnos(alumnoId),  
FOREIGN KEY (citaId) REFERENCES Citas(citaId)  
  
);
```

```
CREATE TABLE CitaTutoria(  
citaTutoriaId INT NOT NULL AUTO_INCREMENT ,  
citaId INT NOT NULL,  
tutoriaId INT NOT NULL,  
PRIMARY KEY (citaTutoriaId),  
FOREIGN KEY (citaId) REFERENCES Citas(citaId),
```

```
FOREIGN KEY (tutoriaId) REFERENCES Tutorias(tutoriaId)

);
```

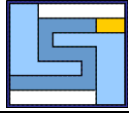
```
CREATE TABLE ProfesorTutoria(
profesorTutoriaId INT NOT NULL AUTO_INCREMENT ,
citaId INT NOT NULL,
tutoriaId INT NOT NULL,
PRIMARY KEY (profesorTutoriaId),
FOREIGN KEY (citaId) REFERENCES Citas(citaId),
FOREIGN KEY (tutoriaId) REFERENCES Tutorias(tutoriaId)
);
```

```
CREATE TABLE DespachoProfesor (
despachoProfesorId INT NOT NULL AUTO_INCREMENT ,
despachoId INT NOT NULL,
profesorId INT NOT NULL,
PRIMARY KEY (despachoProfesorId),
FOREIGN KEY (despachoId) REFERENCES Despachos(despachoId),
FOREIGN KEY (profesorId) REFERENCES Profesores(profesorId)
);
```

```
CREATE TABLE ProfesorDepartamento (
profesorDepartamentoId INT NOT NULL AUTO_INCREMENT ,
profesorId INT NOT NULL,
departamentoId INT NOT NULL,
PRIMARY KEY (profesorDepartamentoId),
FOREIGN KEY (profesorId) REFERENCES Profesores(profesorId),
FOREIGN KEY (departamentoId) REFERENCES Departamentos(departamentoId)
);
```

```
CREATE TABLE DepartamentoAsignatura(
departamentoAsignaturaId INT NOT NULL AUTO_INCREMENT ,
departamentoId INT NOT NULL,
asignaturaId INT NOT NULL,
PRIMARY KEY (departamentoAsignaturaId),
FOREIGN KEY (departamentoId) REFERENCES Departamentos(departamentoId),
FOREIGN KEY (asignaturaId) REFERENCES Asignaturas(asignaturaId)
);
```

```
CREATE TABLE AsignaturaGrado(
asignaturaGradoId INT NOT NULL AUTO_INCREMENT ,
asignaturaId INT NOT NULL,
gradoId INT NOT NULL,
PRIMARY KEY (asignaturaGradoId),
FOREIGN KEY (asignaturaId) REFERENCES Asignaturas(asignaturaId),
```



```
FOREIGN KEY (gradoId) REFERENCES Grados (gradoId)

);
CREATE TABLE GrupoAsignatura (
grupoAsignaturaId INT NOT NULL AUTO_INCREMENT ,
grupoId INT NOT NULL,
asignaturaId INT NOT NULL,
PRIMARY KEY (grupoAsignaturaId),
FOREIGN KEY (grupoId) REFERENCES Grupos (grupoId),
FOREIGN KEY (asignaturaId) REFERENCES Asignaturas (asignaturaId)

);
CREATE TABLE AlumnoNota (
alumnoNotaId INT NOT NULL AUTO_INCREMENT ,
alumnoId INT NOT NULL,
notaId INT NOT NULL,
PRIMARY KEY (alumnoNotaId),
FOREIGN KEY (alumnoId) REFERENCES Alumnos (alumnoId),
FOREIGN KEY (notaId) REFERENCES Notas (notaId)

);

CREATE TABLE NotaGrupo (
notaGrupoId INT NOT NULL AUTO_INCREMENT ,
notaId INT NOT NULL,
grupoId INT NOT NULL,
PRIMARY KEY (notaGrupoId),
FOREIGN KEY (notaId) REFERENCES Notas (notaId),
FOREIGN KEY (grupoId) REFERENCES Grupos (grupoId)

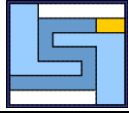
);
```

4.1 Tabla Degrees (Grados)

4.2 Tabla XXX.

```
INSERT INTO Grados (gradosId, nombre, anyos) VALUES
(1, 'Ingeniería del Software', 4),
(2, 'Ingeniería del Computadores', 4),
(3, 'Tecnologías Informáticas', 4);

INSERT INTO Asignaturas (asignaturaId, nombre, acronimo, creditos,
anyo, tipo, gradoId) VALUES
(1, 'Diseño y Pruebas', 'DP', 12, 3, 'Obligatoria', 1),
(2, 'Acceso Inteligente a la Información', 'AII', 6, 4, 'Optativa',
1),
(3, 'Optimización de Sistemas', 'OS', 6, 4, 'Optativa', 1),
(4, 'Ingeniería de Requisitos', 'IR', 6, 2, 'Obligatoria', 1),
(5, 'Análisis y Diseño de Datos y Algoritmos', 'ADDA', 12, 2,
'Obligatoria', 1),
-- 5/6
(6, 'Introducción a la Matemática Discreta', 'IMD', 6, 1, 'Formación
Básica', 2),
(7, 'Redes de Computadores', 'RC', 6, 2, 'Obligatoria', 2),
```

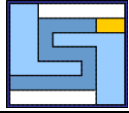


```
(8, 'Teoría de Grafos', 'TG', 6, 3, 'Obligatoria', 2),  
(9, 'Aplicaciones de Soft Computing', 'ASC', 6, 4, 'Optativa', 2),  
-- 9/10  
(10, 'Fundamentos de Programación', 'FP', 12, 1, 'Formacion Basica',  
3),  
(11, 'Lógica Informatica', 'LI', 6, 2, 'Optativa', 3),  
(12, 'Gestión y Estrategia Empresarial', 'GEE', 90, 3, 'Optativa', 3),  
(13, 'Trabajo de Fin de Grado', 'TFG', 12, 4, 'Obligatoria', 3);
```

```
INSERT INTO Grupos (grupoId, nombre, actividad, anyo, asignaturaId)  
VALUES
```

```
(1, 'T1', 'Teoria', 2018, 1),  
(2, 'T2', 'Teoria', 2018, 1),  
(3, 'L1', 'Laboratorio', 2018, 1),  
(4, 'L2', 'Laboratorio', 2018, 1),  
(5, 'L3', 'Laboratorio', 2018, 1),  
(6, 'T1', 'Teoria', 2019, 1),  
(7, 'T2', 'Teoria', 2019, 1),  
(8, 'L1', 'Laboratorio', 2019, 1),  
(9, 'L2', 'Laboratorio', 2019, 1),  
(10, 'Teor1', 'Teoria', 2018, 2),  
(11, 'Teor2', 'Teoria', 2018, 2),  
(12, 'Lab1', 'Laboratorio', 2018, 2),  
(13, 'Lab2', 'Laboratorio', 2018, 2),  
(14, 'Teor1', 'Teoria', 2019, 2),  
(15, 'Lab1', 'Laboratorio', 2019, 2),  
(16, 'Lab2', 'Laboratorio', 2019, 2),  
(17, 'T1', 'Teoria', 2019, 10),  
(18, 'T2', 'Teoria', 2019, 10),  
(19, 'T3', 'Teoria', 2019, 10),  
(20, 'L1', 'Laboratorio', 2019, 10),  
(21, 'L2', 'Laboratorio', 2019, 10),  
(22, 'L3', 'Laboratorio', 2019, 10),  
(23, 'L4', 'Laboratorio', 2019, 10),  
(24, 'Clase', 'Teoria', 2019, 12);
```

```
INSERT INTO Estudiante (estudianteId, metodoAcceso, dni, nombre,  
apellido, fechaNacimiento, email) VALUES  
(1, 'Selectividad', '12345678A', 'Daniel', 'Pérez', '1991-01-01',  
'daniel@alum.us.es'),  
(2, 'Selectividad', '22345678A', 'Rafael', 'Ramírez', '1992-01-01',  
'rafael@alum.us.es'),  
(3, 'Selectividad', '32345678A', 'Gabriel', 'Hernández', '1993-01-01',  
'gabriel@alum.us.es'),  
(4, 'Selectividad', '42345678A', 'Manuel', 'Fernández', '1994-01-01',  
'manuel@alum.us.es'),  
(5, 'Selectividad', '52345678A', 'Joel', 'Gómez', '1995-01-01',  
'joel@alum.us.es'),  
(6, 'Selectividad', '62345678A', 'Abel', 'López', '1996-01-01',  
'abel@alum.us.es'),  
(7, 'Selectividad', '72345678A', 'Azael', 'González', '1997-01-01',  
'azael@alum.us.es'),  
(8, 'Selectividad', '8345678A', 'Uriel', 'Martínez', '1998-01-01',  
'uriel@alum.us.es'),  
(9, 'Selectividad', '92345678A', 'Gael', 'Sánchez', '1999-01-01',  
'gael@alum.us.es'),  
(10, 'Titulado Extranjero', '12345678B', 'Noel', 'Álvarez', '1991-02-  
02', 'noel@alum.us.es'),  
(11, 'Titulado Extranjero', '22345678B', 'Ismael', 'Antúnez', '1992-  
02-02', 'ismael@alum.us.es');
```



```
(12, 'Titulado Extranjero', '32345678B', 'Nathanael', 'Antolinez',  
'1993-02-02', 'nathanael@alum.us.es'),  
(13, 'Titulado Extranjero', '42345678B', 'Ezequiel', 'Aznárez', '1994-  
02-02', 'ezequiel@alum.us.es'),  
(14, 'Titulado Extranjero', '52345678B', 'Ángel', 'Chávez', '1995-02-  
02', 'angel@alum.us.es'),  
(15, 'Titulado Extranjero', '62345678B', 'Matusael', 'Gutiérrez',  
'1996-02-02', 'matusael@alum.us.es'),  
(16, 'Titulado Extranjero', '72345678B', 'Samael', 'Gálvez', '1997-02-  
02', 'samael@alum.us.es'),  
(17, 'Titulado Extranjero', '82345678B', 'Baraquiél', 'Ibáñez', '1998-  
02-02', 'baraquiél@alum.us.es'),  
(18, 'Titulado Extranjero', '92345678B', 'Otoniel', 'Idiáquez', '1999-  
02-02', 'otoniel@alum.us.es'),  
(19, 'Titulado Extranjero', '12345678C', 'Niriel', 'Benítez', '1991-  
03-03', 'niriel@alum.us.es'),  
(20, 'Titulado Extranjero', '22345678C', 'Múriel', 'Bermúdez', '1992-  
03-03', 'muriel@alum.us.es'),  
(21, 'Titulado Extranjero', '32345678C', 'John', 'AII', '2000-01-01',  
'john@alum.us.es');
```

```
INSERT INTO GrupoEstudiantes (GrupoestudianteId, grupoId,  
estudianteId) VALUES
```

```
(1, 1, 1),  
(2, 3, 1),  
(3, 7, 1),  
(4, 8, 1),  
(5, 10, 1),  
(6, 12, 1),  
-- 6/7  
(7, 2, 2),  
(8, 3, 2),  
(9, 10, 2),  
(10, 12, 2),  
-- 10/11  
(11, 18, 21),  
(12, 21, 21);
```

```
INSERT INTO Notas (notaId, puntuacion, gradeCall, mencionHonor,  
estudianteId, grupoId) VALUES
```

```
(1, 4.50, 1, 0, 1, 1),  
(2, 3.25, 2, 0, 1, 1),  
(3, 9.95, 1, 0, 1, 7),  
(4, 7.5, 1, 0, 1, 10),  
-- 4/5  
(5, 2.50, 1, 0, 2, 2),  
(6, 5.00, 2, 0, 2, 2),  
(7, 10.00, 1, 1, 2, 10),  
-- 7/8  
(8, 0.00, 1, 0, 21, 18),  
(9, 1.25, 2, 0, 21, 18),  
(10, 0.5, 3, 0, 21, 18);
```

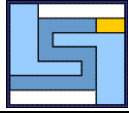
```
INSERT INTO
```

```
Aulas (AulaId, nombre, megafonia, proyector, teoria, laboratorio) VALUES
```

```
(1, Fisica, TRUE, FALSE, TRUE, FALSE),  
(2, biologia, false, FALSE, TRUE, FALSE);
```

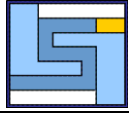
```
INSERT INTO GruposAlumnos (grupoalumnoId, grupoId, alumnoId) VALUES
```

```
(1, 3, 4),  
(2, 1, 2);
```



```
INSERT INTO Tutorias(tutoriaId, profesorId, dia, horaComienzo, horaFin
) VALUES
(1,2,11/12/2020,8:00,9:00),
(2,1,12/1/2020,9:00,9:30);
INSERT INTO Citas(citaId, tutoriaId, alumnoId, hora, fecha) VALUES
(1,2,2,8:00,2/3/2020),
(3,1,1,9:00,1/3/2020)

INSERT INTO Imparte(ImparteId, creditos, profesorId, grupoid) VALUES
(1,6.0,1,1),
(2,12.0,2,1);
INSERT INTO AlumnoCita(alumnoCitaId, alumnoId, citaId) VALUES
(1,2,1),
(2,1,1);
INSERT INTO CitaTutoria(citaTutoriaId, citaId ,tutoriaId) VALUES
(1,2,2),
(2,3,2);
INSERT INTO ProfesorTutoria(profesorTutoriaId ,citaId ,tutoriaId )
VALUES
(1,2,1),
(2,1,1);
INSERT INTO DespachoProfesor (despachoProfesorId despachoId
,profesorId ) VALUES
(1,2,1),
(2,1,1);
INSERT INTO ProfesorDepartamento (profesorDepartamentoId ,profesorId
,departamentoId) VALUES
(1,2,1),
(2,1,1);
INSERT INTO DepartamentoAsignatura(departamentoAsignaturaId
,departamentoId ,asignaturaId ) VALUES
(1,2,1),
(2,1,1);
INSERT INTO AsignaturaGrado(asignaturaGradoId ,asignaturaId ,gradoId
) VALUES
(1,2,1),
(2,1,1);
INSERT INTO GrupoAsignatura(grupoAsignaturaId ,grupoId ,asignaturaId
) VALUES
(1,2,1),
(2,1,1);
INSERT INTO AlumnoNota (alumnoNotaId ,alumnoId ,notaId) VALUES
(1,2,1),
(2,1,1);
INSERT INTO NotaGrupo (notaGrupoId ,notaId ,grupoId )
(1,2,1),
(2,1,1);
```



2.2 Requisitos funcionales.

2.2.1 RF-001. Añadir nota

```
DELIMITER //
```

```
CREATE OR REPLACE PROCEDURE procedimientoAnyadirNota (valor INT,  
convocatoria INT, conMatricula BOOL, alumnoId INT, grupoId INT)  
BEGIN  
INSERT INTO Notas (valor, convocatoria, conMatricula, alumnoId,  
grupoId)  
VALUES ( valor, convocatoria, conMatricula, alumnoId, grupoId);  
END //
```

```
DELIMITER ;
```

2.2.2 RF-002. Citas de una tutoría

```
DELIMITER //
```

```
CREATE OR REPLACE TRIGGER triggerWithHonours  
BEFORE INSERT ON Grades  
FOR EACH ROW  
BEGIN  
IF (new.withHonours = 1 AND new.value < 9.0) THEN  
SIGNAL SQLSTATE '45000' SET message_text =  
'You cannot insert a grade with honours whose value is less than 9';  
END IF;  
END//
```

```
DELIMITER ;
```

2.2.3 RF-003. Asignación de asignaturas

```
DELIMITER //
```

```
CREATE OR REPLACE PROCEDURE procedimientoAsignaturaProfesor  
(profesorId int, grupoId int, creditos INT)  
BEGIN  
INSERT INTO Imparte (grupoId, profesorId, creditos)  
VALUES (grupoId, profesorId, creditos);  
END //
```

```
DELIMITER ;
```

2.2.4 RF-005 Profesor que imparte más créditos en una asignatura

```
DELIMITER //
```

```
CREATE OR REPLACE PROCEDURE procedimientoAsignaturaProfesor  
(profesorId int, grupoId int, creditos INT)  
BEGIN  
INSERT INTO Imparte (grupoId, profesorId, creditos)  
VALUES (grupoId, profesorId, creditos);  
END //
```

```
DELIMITER ;
```

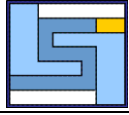
2.2.5 RF-006. Borrar notas de un alumno

```
DELIMITER //
```

```
CREATE OR REPLACE PROCEDURE procDeleteGrades(studentDni CHAR(9))  
BEGIN  
DECLARE id INT;  
SET id = (SELECT studentId FROM Students WHERE dni=studentDni);  
DELETE FROM Grades WHERE studentId=id;  
END //
```

```
DELIMITER ;
```

```
DELIMITER //
```



```
CREATE OR REPLACE PROCEDURE procDeleteData ()
BEGIN
DELETE FROM Grades;
DELETE FROM GroupsStudents;
DELETE FROM Students;
DELETE FROM Groups;
DELETE FROM Subjects;
DELETE FROM Degrees;
END //
DELIMITER ;

-- CALL procDeleteGrades('12345678A');
CALL procDeleteData();

DELIMITER //
CREATE OR REPLACE FUNCTION avgGrade(studentId INT) RETURNS DOUBLE
BEGIN
DECLARE avgStudentGrade DOUBLE;
SET avgStudentGrade = (SELECT AVG(value) FROM Grades
WHERE Grades.studentId = studentId);
RETURN avgStudentGrade;
END //
DELIMITER ;
```

2.3.3 RN-007 Nota por alumno

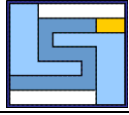
```
DELIMITER //
CREATE OR REPLACE PROCEDURE procedimientoAnyadirNota (valor INT,
convocatoria INT, conMatricula BOOL, alumnoId INT, grupoId INT)
BEGIN
INSERT INTO Notas (valor, convocatoria, conMatricula, alumnoId,
grupoId)
VALUES ( valor, convocatoria, conMatricula, alumnoId, grupoId);
END //
DELIMITER ;
```

2.2.6 RF-011. Nota media de alumno

```
DELIMITER //
CREATE OR REPLACE FUNCTION avgGrade(studentId INT) RETURNS DOUBLE
BEGIN
DECLARE avgStudentGrade DOUBLE;
SET avgStudentGrade = (SELECT AVG(value) FROM Grades
WHERE Grades.studentId = studentId);
RETURN avgStudentGrade;
END //
DELIMITER ;
```

2.2.7 RF-036 Listado de alumnos de una asignatura por grupos

```
DELIMITER //
CREATE OR REPLACE VIEW verListaAlumnos AS
SELECT Alumnos.firstNombre, Alumnos.surNombre, Asignatura.name,
grupos.grupoId FROM Alumnos
JOIN GrupoAlumnos ON (Alumnos.alumnoId = grupoAlumnos.alumnoId)
JOIN grupos ON (grupoAlumnos.grupoId = grupos.grupoId)
JOIN Asignatura ON (Asignatura.asignaturaId = grupos.asignaturaId)
JOIN grados ON (Asignatura.gradoId=Grados.gradoId) ;
SELECT * FROM verListaAlumnos;
```



DELIMITER ;

2.2.8 RF-037 Listado de grupos por profesor y asignatura

DELIMITER //

```
CREATE OR REPLACE FUNCTION funcionalumnoprofesor (alumId INT ) RETURNS  
DOUBLE  
BEGIN  
RETURN (SELECT AVG(value) FROM notas WHERE alumnoId=alumId);  
  
END //
```

DELIMITER ;

2.2.9 RF-038 Listado de despachos con más de un profesor

DELIMITER //

```
CREATE OR REPLACE VIEW maslprofesor AS  
SELECT * FROM profesores WHERE profesores => 1;  
  
SELECT * FROM maslprofesor;
```

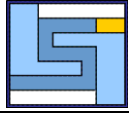
DELIMITER ;

2.2.10 RF-039 Listado de tutorías pendientes de cada profesor

DELIMITER //

```
CREATE OR REPLACE FUNCTION funciotutorias (profesorId INT ) RETURNS  
DOUBLE  
BEGIN  
RETURN (SELECT AVG(value) FROM notas WHERE tutoriaId= tutoriaId);  
  
END //
```

DELIMITER ;



2.3 Reglas de negocio.

2.3.1 RN-001 Limitación créditos docentes

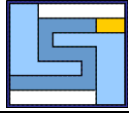
```
CREATE OR REPLACE TRIGGER triggerMaximumCredits
BEFORE INSERT ON GroupsStudents
FOR EACH ROW
BEGIN
    DECLARE subject INT;
    DECLARE groupYear INT;
    DECLARE alreadyBelongs BOOLEAN;
    DECLARE newCredits INT;
    DECLARE currentCredits INT;
    SELECT subjectId, year INTO subject, groupYear FROM Groups WHERE
    groupId = new.groupId;

    -- Vamos si la asignatura del nuevo grupo es una a la que ya
    pertenecía este año
    SET already_belongs = (SELECT COUNT(DISTINCT(subjectId))
    FROM ViewStudentSubjects
    WHERE (studentId = new.studentId AND
    subjectId = subject AND
    year = groupYear));
    -- Solo es necesario seguir comprobando si es una asignatura nueva
    IF (NOT alreadyBelongs) THEN
        -- Obtenemos los créditos de la asignatura nueva
        SET newCredits = (SELECT credits FROM Subjects WHERE subjectId =
        subject);
        -- Obtenemos los créditos de las asignaturas en las que estaba
        SET currentCredits = (SELECT SUM(credits)
        FROM Subjects
        WHERE Subjects.subjectId IN (
        SELECT Groups.subjectId
        FROM Groups
        INNER JOIN GroupsStudents ON (Groups.groupId=GroupsStudents.groupId)
        WHERE (studentId=new.studentId AND
        year = groupYear)));
        IF ((currentCredits + newCredits) > 90) THEN
            SIGNAL SQLSTATE '45000' SET message_text =
            'Un alumno no puede tener asignaturas con más de 90 créditos un mismo
            año';
        END IF;
    END IF;
END//

DELIMITER ;
```

2.3.2 RN-006 Matrícula de honor

```
DELIMITER //
CREATE OR REPLACE PROCEDURE procDeleteGrades(studentDni CHAR(9))
BEGIN
    DECLARE id INT;
    SET id = (SELECT studentId FROM Students WHERE dni=studentDni);
    DELETE FROM Grades WHERE studentId=id;
END //
DELIMITER ;
```



```
DELIMITER //
```

```
CREATE OR REPLACE PROCEDURE procDeleteData()  
BEGIN  
DELETE FROM Grades;  
DELETE FROM GroupsStudents;  
DELETE FROM Students;  
DELETE FROM Groups;  
DELETE FROM Subjects;  
DELETE FROM Degrees;  
END //
```

```
DELIMITER ;
```

```
-- CALL procDeleteGrades('12345678A');  
CALL procDeleteData();
```

```
DELIMITER //
```

```
CREATE OR REPLACE FUNCTION avgGrade(studentId INT) RETURNS DOUBLE  
BEGIN  
DECLARE avgStudentGrade DOUBLE;  
SET avgStudentGrade = (SELECT AVG(value) FROM Grades  
WHERE Grades.studentId = studentId);  
RETURN avgStudentGrade;  
END //
```

```
DELIMITER ;
```

2.3.3 RN-007 Nota por alumno

```
DELIMITER //
```

```
CREATE OR REPLACE PROCEDURE procedimientoAnyadirNota (valor INT,  
convocatoria INT, conMatricula BOOL, alumnoId INT, grupoId INT)  
BEGIN  
INSERT INTO Notas (valor, convocatoria, conMatricula, alumnoId,  
grupoId)  
VALUES ( valor, convocatoria, conMatricula, alumnoId, grupoId);  
END //
```

```
DELIMITER ;
```

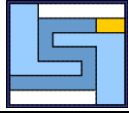
2.3.4 RN-010 Cambios bruscos en notas

```
DELIMITER//
```

```
CREATE OR REPLACE TRIGGER triggerGradesChangesDiference  
BEFORE INSERT ON Grades  
FOR EACH ROW  
BEGIN  
DECLARE difference DECIMAL(4,2);  
DECLARE student ROW TYPE OF Students;  
SET difference=new.value-old.value;
```

```
if(difference>4) THEN  
SELECT * INTO student FROM Students WHERE studentId=new.studentId;  
SET @error message=CONCAT("al alumno", student.firstName, " ",  
student.surname,"se intento subir una nota en" difference,"puntos");  
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT=  
error message  
END if;
```

```
END//
```



2.3.5 RN-019 Alumno pertenece a grupo

```
CREATE OR REPLACE TRIGGER triggerNotasGrupoEstudiantes
BEFORE INSERT ON Notas
FOR EACH ROW
BEGIN
DECLARE estaEnGrupo INT;
SET estaEnGrupo=(SELECT COUNT(*)
FROM GruposAlumnos
WHERE alumnoId=nuevo.alumnoId AND grupoId=nuevo.grupoId);
if(estaEnGrupo< 1) then
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT=
'Un alumno no puede tener notas en grupos a los que no pertenece';
END if;
END//
```

2.3.6 RN-020 Profesor de teoría y prácticas

```
DELIMITER //
CREATE OR REPLACE TRIGGER triggerprofesorTEoriaPracticas
BEFORE INSERT ON Notas
FOR EACH ROW
BEGIN
DECLARE estaEnGrupo INT;
SET estaEnGrupo=(SELECT COUNT(*)
FROM GruposAlumnos
WHERE alumnoId=nuevo.alumnoId AND grupoId=nuevo.grupoId);
if(estaEnGrupo< 1) then
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT=
'Un alumno no puede tener notas en grupos a los que no pertenece';
END if;
END//

DELIMITER;
```

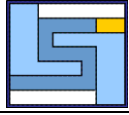
2.3.7 RN-021 Aulas de teoría y prácticas

```
Código del disparador para DELIMITER//
CREATE OR REPLACE TRIGGER triggerGradesChangesDiferrence
BEFORE INSERT ON Grades
FOR EACH ROW
BEGIN
DECLARE diferrence DECIMAL(4,2);
DECLARE student ROW TYPE OF Students;
SET diferrence=new.value=old.value;

if(diferrence>4) THEN
SELECT * INTO student FROM Students WHERE studentId=new.studentId;
SET @error message=CONCAT("al alumno", student.firstName," ",
student.surname,"se intento subir una nota en" diferrence,"puntos");
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT=
error message
END if;
END//
```

2.3.8 RN-022 Citas para tutorías

```
DELIMITER//
CREATE OR REPLACE TRIGGER triggerAppointments (dWeek VARCHAR(30),
```



```
idT INT)
```

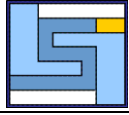
```
BEGIN
```

```
SELECT * FROM viewTutoiraApointemts NATURAL JOIN Students WHERE  
teahcerID=idT AND dayWeek= dWeek;
```

```
END if;
```

```
END//
```

```
DELIMITER ;
```



3. Anexo con todo el código SQL.

GIT: <https://github.com/josvilcar3/ISSI-1-E3-Josvicar3>

```
-----
-- Host:                                127.0.0.1
-- Versión del servidor:                 10.4.8-MariaDB - mariadb.org binary
distribution
-- SO del servidor:                     Win64
-- HeidiSQL Versión:                    10.2.0.5599
-----

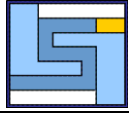
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET NAMES utf8 */;
/*!50503 SET NAMES utf8mb4 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

-- Volcando estructura de base de datos para grados
CREATE DATABASE IF NOT EXISTS `grados` /*!40100 DEFAULT CHARACTER SET
latin1 COLLATE latin1_spanish_ci */;
USE `grados`;

-- Volcando estructura para tabla grados.alumnocita
DROP TABLE IF EXISTS `alumnocita`;
CREATE TABLE IF NOT EXISTS `alumnocita` (
  `alumnoCitaId` int(11) NOT NULL AUTO_INCREMENT,
  `alumnoId` int(11) NOT NULL,
  `citaId` int(11) NOT NULL,
  PRIMARY KEY (`alumnoCitaId`),
  KEY `alumnoId` (`alumnoId`),
  KEY `citaId` (`citaId`),
  CONSTRAINT `alumnocita_ibfk_1` FOREIGN KEY (`alumnoId`) REFERENCES
`alumnos` (`alumnoId`),
  CONSTRAINT `alumnocita_ibfk_2` FOREIGN KEY (`citaId`) REFERENCES
`citas` (`citaId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.alumnocita: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `alumnocita` DISABLE KEYS */;
/*!40000 ALTER TABLE `alumnocita` ENABLE KEYS */;

-- Volcando estructura para tabla grados.alumnonota
DROP TABLE IF EXISTS `alumnonota`;
CREATE TABLE IF NOT EXISTS `alumnonota` (
  `alumnoNotaId` int(11) NOT NULL AUTO_INCREMENT,
  `alumnoId` int(11) NOT NULL,
  `notaId` int(11) NOT NULL,
  PRIMARY KEY (`alumnoNotaId`),
  KEY `alumnoId` (`alumnoId`),
```



```
KEY `notaId` (`notaId`),
CONSTRAINT `alumnonota_ibfk_1` FOREIGN KEY (`alumnoId`) REFERENCES
`alumnos` (`alumnoId`),
CONSTRAINT `alumnonota_ibfk_2` FOREIGN KEY (`notaId`) REFERENCES
`notas` (`notaId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.alumnonota: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `alumnonota` DISABLE KEYS */;
/*!40000 ALTER TABLE `alumnonota` ENABLE KEYS */;

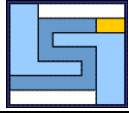
-- Volcando estructura para tabla grados.alumnos
DROP TABLE IF EXISTS `alumnos`;
CREATE TABLE IF NOT EXISTS `alumnos` (
  `alumnoId` int(11) NOT NULL AUTO_INCREMENT,
  `metodoAcceso` varchar(30) COLLATE latin1_spanish_ci NOT NULL,
  `dni` char(9) COLLATE latin1_spanish_ci NOT NULL,
  `firstNombre` varchar(100) COLLATE latin1_spanish_ci NOT NULL,
  `surNombre` varchar(100) COLLATE latin1_spanish_ci NOT NULL,
  `birthDate` date NOT NULL,
  `email` varchar(250) COLLATE latin1_spanish_ci NOT NULL,
  PRIMARY KEY (`alumnoId`),
  UNIQUE KEY `dni` (`dni`),
  CONSTRAINT `invalidStudentmetodoAcceso` CHECK (`metodoAcceso` in
('Selectividad','Ciclo','Mayor','Titulado Extranjero'))
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.alumnos: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `alumnos` DISABLE KEYS */;
/*!40000 ALTER TABLE `alumnos` ENABLE KEYS */;

-- Volcando estructura para tabla grados.asignaturagrado
DROP TABLE IF EXISTS `asignaturagrado`;
CREATE TABLE IF NOT EXISTS `asignaturagrado` (
  `asignaturaGradoId` int(11) NOT NULL AUTO_INCREMENT,
  `asignaturaId` int(11) NOT NULL,
  `gradoId` int(11) NOT NULL,
  PRIMARY KEY (`asignaturaGradoId`),
  KEY `asignaturaId` (`asignaturaId`),
  KEY `gradoId` (`gradoId`),
  CONSTRAINT `asignaturagrado_ibfk_1` FOREIGN KEY (`asignaturaId`)
REFERENCES `asignaturas` (`asignaturaId`),
  CONSTRAINT `asignaturagrado_ibfk_2` FOREIGN KEY (`gradoId`) REFERENCES
`grados` (`gradoId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.asignaturagrado: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `asignaturagrado` DISABLE KEYS */;
/*!40000 ALTER TABLE `asignaturagrado` ENABLE KEYS */;

-- Volcando estructura para tabla grados.asignaturas
DROP TABLE IF EXISTS `asignaturas`;
CREATE TABLE IF NOT EXISTS `asignaturas` (
  `asignaturaId` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(100) COLLATE latin1_spanish_ci NOT NULL,
  `acronimo` varchar(8) COLLATE latin1_spanish_ci NOT NULL,
  `creditos` int(11) NOT NULL,
```



```
`curso` int(11) NOT NULL,
`tipo` varchar(20) COLLATE latin1_spanish_ci NOT NULL,
`gradoId` int(11) NOT NULL,
PRIMARY KEY (`asignaturaId`),
UNIQUE KEY `nombre` (`nombre`),
UNIQUE KEY `acronimo` (`acronimo`),
KEY `gradoId` (`gradoId`),
CONSTRAINT `asignaturas_ibfk_1` FOREIGN KEY (`gradoId`) REFERENCES
`grados` (`gradoId`),
CONSTRAINT `negativeSubjectcredits` CHECK (`credits` > 0),
CONSTRAINT `invalidSubjectcurso` CHECK (`curso` > 0 and `curso` < 6),
CONSTRAINT `invalidSubjecttipo` CHECK (`tipo` in ('Formacion
Basica','Optativa','Obligatoria'))
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.asignaturas: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `asignaturas` DISABLE KEYS */;
/*!40000 ALTER TABLE `asignaturas` ENABLE KEYS */;

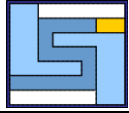
-- Volcando estructura para tabla grados.aulagrupo
DROP TABLE IF EXISTS `aulagrupo`;
CREATE TABLE IF NOT EXISTS `aulagrupo` (
`aulaGrupoId` int(11) NOT NULL AUTO_INCREMENT,
`aulaId` int(11) NOT NULL,
`grupoId` int(11) NOT NULL,
PRIMARY KEY (`aulaGrupoId`),
KEY `aulaId` (`aulaId`),
KEY `grupoId` (`grupoId`),
CONSTRAINT `aulagrupo_ibfk_1` FOREIGN KEY (`aulaId`) REFERENCES
`aulas` (`aulaId`),
CONSTRAINT `aulagrupo_ibfk_2` FOREIGN KEY (`grupoId`) REFERENCES
`grupos` (`grupoId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.aulagrupo: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `aulagrupo` DISABLE KEYS */;
/*!40000 ALTER TABLE `aulagrupo` ENABLE KEYS */;

-- Volcando estructura para tabla grados.aulas
DROP TABLE IF EXISTS `aulas`;
CREATE TABLE IF NOT EXISTS `aulas` (
`aulaId` int(11) NOT NULL AUTO_INCREMENT,
`nombre` varchar(100) COLLATE latin1_spanish_ci NOT NULL,
`planta` int(11) NOT NULL,
`capacidad` int(11) NOT NULL,
`tieneProyector` tinyint(1) NOT NULL,
`tieneAltavoces` tinyint(1) NOT NULL,
PRIMARY KEY (`aulaId`),
UNIQUE KEY `nombre` (`nombre`),
UNIQUE KEY `planta` (`planta`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.aulas: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `aulas` DISABLE KEYS */;
/*!40000 ALTER TABLE `aulas` ENABLE KEYS */;

-- Volcando estructura para tabla grados.citas
```



```
DROP TABLE IF EXISTS `citas`;
CREATE TABLE IF NOT EXISTS `citas` (
  `citaId` int(11) NOT NULL AUTO_INCREMENT,
  `tutoriaId` int(11) NOT NULL,
  `alumnoId` int(11) NOT NULL,
  `hora` time NOT NULL,
  `fecha` date NOT NULL,
  PRIMARY KEY (`citaId`),
  KEY `tutoriaId` (`tutoriaId`),
  KEY `alumnoId` (`alumnoId`),
  CONSTRAINT `citas_ibfk_1` FOREIGN KEY (`tutoriaId`) REFERENCES
  `tutorias` (`tutoriaId`),
  CONSTRAINT `citas_ibfk_2` FOREIGN KEY (`alumnoId`) REFERENCES
  `alumnos` (`alumnoId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

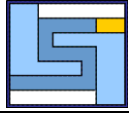
-- Volcando datos para la tabla grados.citas: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `citas` DISABLE KEYS */;
/*!40000 ALTER TABLE `citas` ENABLE KEYS */;

-- Volcando estructura para tabla grados.citatutoria
DROP TABLE IF EXISTS `citaturia`;
CREATE TABLE IF NOT EXISTS `citaturia` (
  `citaTutoriaId` int(11) NOT NULL AUTO_INCREMENT,
  `citaId` int(11) NOT NULL,
  `tutoriaId` int(11) NOT NULL,
  PRIMARY KEY (`citaTutoriaId`),
  KEY `citaId` (`citaId`),
  KEY `tutoriaId` (`tutoriaId`),
  CONSTRAINT `citaturia_ibfk_1` FOREIGN KEY (`citaId`) REFERENCES
  `citas` (`citaId`),
  CONSTRAINT `citaturia_ibfk_2` FOREIGN KEY (`tutoriaId`) REFERENCES
  `tutorias` (`tutoriaId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.citatutoria: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `citaturia` DISABLE KEYS */;
/*!40000 ALTER TABLE `citaturia` ENABLE KEYS */;

-- Volcando estructura para tabla grados.degrees
DROP TABLE IF EXISTS `degrees`;
CREATE TABLE IF NOT EXISTS `degrees` (
  `degreeId` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(60) COLLATE latin1_spanish_ci NOT NULL,
  `years` int(11) NOT NULL DEFAULT 4,
  PRIMARY KEY (`degreeId`),
  UNIQUE KEY `name` (`name`),
  CONSTRAINT `invalidDegreeYear` CHECK (`years` >= 3 and `years` <= 5)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1
COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.degrees: ~3 rows
(aproximadamente)
/*!40000 ALTER TABLE `degrees` DISABLE KEYS */;
REPLACE INTO `degrees` (`degreeId`, `name`, `years`) VALUES
(1, 'Ingeniería del Software', 4),
(2, 'Ingeniería del Computadores', 4),
(3, 'Tecnologías Informáticas', 4);
```

```
/*!40000 ALTER TABLE `degrees` ENABLE KEYS */;

-- Volcando estructura para tabla grados.departamentoasignatura
DROP TABLE IF EXISTS `departamentoasignatura`;
CREATE TABLE IF NOT EXISTS `departamentoasignatura` (
  `departamentoAsignaturaId` int(11) NOT NULL AUTO_INCREMENT,
  `departamentoId` int(11) NOT NULL,
  `asignaturaId` int(11) NOT NULL,
  PRIMARY KEY (`departamentoAsignaturaId`),
  KEY `departamentoId` (`departamentoId`),
  KEY `asignaturaId` (`asignaturaId`),
  CONSTRAINT `departamentoasignatura_ibfk_1` FOREIGN KEY
    (`departamentoId`) REFERENCES `departamentos` (`departamentoId`),
  CONSTRAINT `departamentoasignatura_ibfk_2` FOREIGN KEY
    (`asignaturaId`) REFERENCES `asignaturas` (`asignaturaId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.departamentoasignatura: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `departamentoasignatura` DISABLE KEYS */;
/*!40000 ALTER TABLE `departamentoasignatura` ENABLE KEYS */;

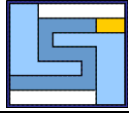
-- Volcando estructura para tabla grados.departamentos
DROP TABLE IF EXISTS `departamentos`;
CREATE TABLE IF NOT EXISTS `departamentos` (
  `departamentoId` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(100) COLLATE latin1_spanish_ci NOT NULL,
  PRIMARY KEY (`departamentoId`),
  UNIQUE KEY `nombre` (`nombre`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.departamentos: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `departamentos` DISABLE KEYS */;
/*!40000 ALTER TABLE `departamentos` ENABLE KEYS */;

-- Volcando estructura para tabla grados.despachoprofesor
DROP TABLE IF EXISTS `despachoprofesor`;
CREATE TABLE IF NOT EXISTS `despachoprofesor` (
  `despachoProfesorId` int(11) NOT NULL AUTO_INCREMENT,
  `despachoId` int(11) NOT NULL,
  `profesorId` int(11) NOT NULL,
  PRIMARY KEY (`despachoProfesorId`),
  KEY `despachoId` (`despachoId`),
  KEY `profesorId` (`profesorId`),
  CONSTRAINT `despachoprofesor_ibfk_1` FOREIGN KEY (`despachoId`)
    REFERENCES `despachos` (`despachoId`),
  CONSTRAINT `despachoprofesor_ibfk_2` FOREIGN KEY (`profesorId`)
    REFERENCES `profesores` (`profesorId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.despachoprofesor: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `despachoprofesor` DISABLE KEYS */;
/*!40000 ALTER TABLE `despachoprofesor` ENABLE KEYS */;

-- Volcando estructura para tabla grados.despachos
DROP TABLE IF EXISTS `despachos`;
CREATE TABLE IF NOT EXISTS `despachos` (
  `despachoId` int(11) NOT NULL AUTO_INCREMENT,
```



```
`nombre` varchar(100) COLLATE latin1_spanish_ci NOT NULL,  
`Planta` int(11) NOT NULL,  
PRIMARY KEY (`despachoId`),  
UNIQUE KEY `nombre` (`nombre`),  
UNIQUE KEY `Planta` (`Planta`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;
```

```
-- Volcando datos para la tabla grados.despachos: ~0 rows  
(aproximadamente)
```

```
/*!40000 ALTER TABLE `despachos` DISABLE KEYS */;  
/*!40000 ALTER TABLE `despachos` ENABLE KEYS */;
```

```
-- Volcando estructura para tabla grados.espaciosaulas
```

```
DROP TABLE IF EXISTS `espaciosaulas`;  
CREATE TABLE IF NOT EXISTS `espaciosaulas` (  
  `aulaId` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre` varchar(100) COLLATE latin1_spanish_ci NOT NULL,  
  `planta` int(11) NOT NULL,  
  `capacidad` int(11) NOT NULL,  
  `tieneProyector` tinyint(1) NOT NULL,  
  `tieneAltavoces` tinyint(1) NOT NULL,  
  PRIMARY KEY (`aulaId`),  
  UNIQUE KEY `nombre` (`nombre`),  
  UNIQUE KEY `planta` (`planta`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;
```

```
-- Volcando datos para la tabla grados.espaciosaulas: ~0 rows  
(aproximadamente)
```

```
/*!40000 ALTER TABLE `espaciosaulas` DISABLE KEYS */;  
/*!40000 ALTER TABLE `espaciosaulas` ENABLE KEYS */;
```

```
-- Volcando estructura para tabla grados.espaciosdespachos
```

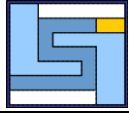
```
DROP TABLE IF EXISTS `espaciosdespachos`;  
CREATE TABLE IF NOT EXISTS `espaciosdespachos` (  
  `despachoId` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre` varchar(100) COLLATE latin1_spanish_ci NOT NULL,  
  `Planta` int(11) NOT NULL,  
  PRIMARY KEY (`despachoId`),  
  UNIQUE KEY `nombre` (`nombre`),  
  UNIQUE KEY `Planta` (`Planta`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;
```

```
-- Volcando datos para la tabla grados.espaciosdespachos: ~0 rows  
(aproximadamente)
```

```
/*!40000 ALTER TABLE `espaciosdespachos` DISABLE KEYS */;  
/*!40000 ALTER TABLE `espaciosdespachos` ENABLE KEYS */;
```

```
-- Volcando estructura para tabla grados.estudiantes
```

```
DROP TABLE IF EXISTS `estudiantes`;  
CREATE TABLE IF NOT EXISTS `estudiantes` (  
  `estudianteId` int(11) NOT NULL AUTO_INCREMENT,  
  `metodoAcceso` varchar(30) COLLATE latin1_spanish_ci NOT NULL,  
  `dni` char(9) COLLATE latin1_spanish_ci NOT NULL,  
  `firstnombre` varchar(100) COLLATE latin1_spanish_ci NOT NULL,  
  `surnombre` varchar(100) COLLATE latin1_spanish_ci NOT NULL,  
  `birthDate` date NOT NULL,  
  `email` varchar(250) COLLATE latin1_spanish_ci NOT NULL,  
  PRIMARY KEY (`estudianteId`),  
  UNIQUE KEY `dni` (`dni`),
```



```
CONSTRAINT `invalidStudentmetodoAcceso` CHECK (`metodoAcceso` in
('Selectividad','Ciclo','Mayor','Titulado Extranjero'))
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.estudiantes: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `estudiantes` DISABLE KEYS */;
/*!40000 ALTER TABLE `estudiantes` ENABLE KEYS */;

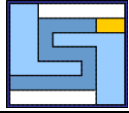
-- Volcando estructura para tabla grados.grados
DROP TABLE IF EXISTS `grados`;
CREATE TABLE IF NOT EXISTS `grados` (
  `gradoId` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(60) COLLATE latin1_spanish_ci NOT NULL,
  `anyos` int(11) NOT NULL DEFAULT 4,
  PRIMARY KEY (`gradoId`),
  UNIQUE KEY `nombre` (`nombre`),
  CONSTRAINT `invalidDegreeanyos` CHECK (`anyos` >= 3 and `anyos` <= 5)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.grados: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `grados` DISABLE KEYS */;
/*!40000 ALTER TABLE `grados` ENABLE KEYS */;

-- Volcando estructura para tabla grados.groups
DROP TABLE IF EXISTS `groups`;
CREATE TABLE IF NOT EXISTS `groups` (
  `groupId` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(30) COLLATE latin1_spanish_ci NOT NULL,
  `activity` varchar(20) COLLATE latin1_spanish_ci NOT NULL,
  `year` int(11) NOT NULL,
  `subjectId` int(11) NOT NULL,
  PRIMARY KEY (`groupId`),
  UNIQUE KEY `name` (`name`,`year`,`subjectId`),
  KEY `subjectId` (`subjectId`),
  CONSTRAINT `groups_ibfk_1` FOREIGN KEY (`subjectId`) REFERENCES
  `subjects` (`subjectId`),
  CONSTRAINT `negativeGroupYear` CHECK (`year` > 0),
  CONSTRAINT `invalidGroupActivity` CHECK (`activity` in
  ('Teoria','Laboratorio'))
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.groups: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `groups` DISABLE KEYS */;
/*!40000 ALTER TABLE `groups` ENABLE KEYS */;

-- Volcando estructura para tabla grados.groupsstudents
DROP TABLE IF EXISTS `groupsstudents`;
CREATE TABLE IF NOT EXISTS `groupsstudents` (
  `groupStudentId` int(11) NOT NULL AUTO_INCREMENT,
  `groupId` int(11) NOT NULL,
  `studentId` int(11) NOT NULL,
  PRIMARY KEY (`groupStudentId`),
  UNIQUE KEY `groupId` (`groupId`,`studentId`),
  KEY `studentId` (`studentId`),
  CONSTRAINT `groupsstudents_ibfk_1` FOREIGN KEY (`groupId`) REFERENCES
  `groups` (`groupId`) ON DELETE CASCADE,
```



```
CONSTRAINT `groupsstudents_ibfk_2` FOREIGN KEY (`studentId`)
REFERENCES `students` (`studentId`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1
COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.groupsstudents: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `groupsstudents` DISABLE KEYS */;
/*!40000 ALTER TABLE `groupsstudents` ENABLE KEYS */;

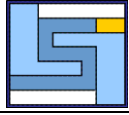
-- Volcando estructura para tabla grados.grupoasignatura
DROP TABLE IF EXISTS `grupoasignatura`;
CREATE TABLE IF NOT EXISTS `grupoasignatura` (
  `grupoAsignaturaId` int(11) NOT NULL AUTO_INCREMENT,
  `grupoId` int(11) NOT NULL,
  `asignaturaId` int(11) NOT NULL,
  PRIMARY KEY (`grupoAsignaturaId`),
  KEY `grupoId` (`grupoId`),
  KEY `asignaturaId` (`asignaturaId`),
  CONSTRAINT `grupoasignatura_ibfk_1` FOREIGN KEY (`grupoId`) REFERENCES
  `grupos` (`grupoId`),
  CONSTRAINT `grupoasignatura_ibfk_2` FOREIGN KEY (`asignaturaId`)
  REFERENCES `asignaturas` (`asignaturaId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.grupoasignatura: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `grupoasignatura` DISABLE KEYS */;
/*!40000 ALTER TABLE `grupoasignatura` ENABLE KEYS */;

-- Volcando estructura para tabla grados.grupos
DROP TABLE IF EXISTS `grupos`;
CREATE TABLE IF NOT EXISTS `grupos` (
  `grupoId` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(30) COLLATE latin1_spanish_ci NOT NULL,
  `actividad` varchar(20) COLLATE latin1_spanish_ci NOT NULL,
  `anyos` int(11) NOT NULL,
  `asignaturaId` int(11) NOT NULL,
  PRIMARY KEY (`grupoId`),
  UNIQUE KEY `nombre` (`nombre`, `anyos`, `asignaturaId`),
  KEY `asignaturaId` (`asignaturaId`),
  CONSTRAINT `grupos_ibfk_1` FOREIGN KEY (`asignaturaId`) REFERENCES
  `asignaturas` (`asignaturaId`),
  CONSTRAINT `negativeGroupanyos` CHECK (`anyos` > 0),
  CONSTRAINT `invalidGroupactividad` CHECK (`actividad` in
  ('Teoria', 'Laboratorio'))
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.grupos: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `grupos` DISABLE KEYS */;
/*!40000 ALTER TABLE `grupos` ENABLE KEYS */;

-- Volcando estructura para tabla grados.gruposalumnos
DROP TABLE IF EXISTS `gruposalumnos`;
CREATE TABLE IF NOT EXISTS `gruposalumnos` (
  `grupoalumnoId` int(11) NOT NULL AUTO_INCREMENT,
  `grupoId` int(11) NOT NULL,
  `alumnoId` int(11) NOT NULL,
  PRIMARY KEY (`grupoalumnoId`),
```



```
UNIQUE KEY `grupoId` (`grupoId`,`alumnoId`),
KEY `alumnoId` (`alumnoId`),
CONSTRAINT `gruposalumnos_ibfk_1` FOREIGN KEY (`grupoId`) REFERENCES
`grupos` (`grupoId`) ON DELETE CASCADE,
CONSTRAINT `gruposalumnos_ibfk_2` FOREIGN KEY (`alumnoId`) REFERENCES
`alumnos` (`alumnoId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.gruposalumnos: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `gruposalumnos` DISABLE KEYS */;
/*!40000 ALTER TABLE `gruposalumnos` ENABLE KEYS */;

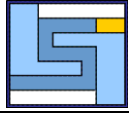
-- Volcando estructura para tabla grados.gruposestudiantes
DROP TABLE IF EXISTS `gruposestudiantes`;
CREATE TABLE IF NOT EXISTS `gruposestudiantes` (
`grupoestudianteId` int(11) NOT NULL AUTO_INCREMENT,
`grupoId` int(11) NOT NULL,
`estudianteId` int(11) NOT NULL,
PRIMARY KEY (`grupoestudianteId`),
UNIQUE KEY `grupoId` (`grupoId`,`estudianteId`),
KEY `estudianteId` (`estudianteId`),
CONSTRAINT `gruposestudiantes_ibfk_1` FOREIGN KEY (`grupoId`)
REFERENCES `grupos` (`grupoId`) ON DELETE CASCADE,
CONSTRAINT `gruposestudiantes_ibfk_2` FOREIGN KEY (`estudianteId`)
REFERENCES `estudiantes` (`estudianteId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.gruposestudiantes: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `gruposestudiantes` DISABLE KEYS */;
/*!40000 ALTER TABLE `gruposestudiantes` ENABLE KEYS */;

-- Volcando estructura para tabla grados.imparte
DROP TABLE IF EXISTS `imparte`;
CREATE TABLE IF NOT EXISTS `imparte` (
`ImparteId` int(11) NOT NULL AUTO_INCREMENT,
`creditos` double NOT NULL,
`profesorId` int(11) NOT NULL,
`grupoid` int(11) NOT NULL,
PRIMARY KEY (`ImparteId`),
KEY `profesorId` (`profesorId`),
KEY `grupoid` (`grupoid`),
CONSTRAINT `imparte_ibfk_1` FOREIGN KEY (`profesorId`) REFERENCES
`profesores` (`profesorId`),
CONSTRAINT `imparte_ibfk_2` FOREIGN KEY (`grupoid`) REFERENCES
`grupos` (`grupoId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.imparte: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `imparte` DISABLE KEYS */;
/*!40000 ALTER TABLE `imparte` ENABLE KEYS */;

-- Volcando estructura para tabla grados.notagrupo
DROP TABLE IF EXISTS `notagrupo`;
CREATE TABLE IF NOT EXISTS `notagrupo` (
`notaGrupoId` int(11) NOT NULL AUTO_INCREMENT,
`notaId` int(11) NOT NULL,
`grupoId` int(11) NOT NULL,
```



```
PRIMARY KEY (`notaGrupoId`),
KEY `notaId` (`notaId`),
KEY `grupoId` (`grupoId`),
CONSTRAINT `notagrupo_ibfk_1` FOREIGN KEY (`notaId`) REFERENCES
`notas` (`notaId`),
CONSTRAINT `notagrupo_ibfk_2` FOREIGN KEY (`grupoId`) REFERENCES
`grupos` (`grupoId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

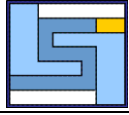
-- Volcando datos para la tabla grados.notagrupo: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `notagrupo` DISABLE KEYS */;
/*!40000 ALTER TABLE `notagrupo` ENABLE KEYS */;

-- Volcando estructura para tabla grados.notas
DROP TABLE IF EXISTS `notas`;
CREATE TABLE IF NOT EXISTS `notas` (
`notaId` int(11) NOT NULL AUTO_INCREMENT,
`valor` decimal(4,2) NOT NULL,
`convocatoria` int(11) NOT NULL,
`conMatricula` tinyint(1) NOT NULL,
`alumnoId` int(11) NOT NULL,
`grupoId` int(11) NOT NULL,
PRIMARY KEY (`notaId`),
UNIQUE KEY `duplicatedCallGrade`
(`convocatoria`,`alumnoId`,`grupoId`),
KEY `alumnoId` (`alumnoId`),
KEY `grupoId` (`grupoId`),
CONSTRAINT `notas_ibfk_1` FOREIGN KEY (`alumnoId`) REFERENCES
`alumnos` (`alumnoId`),
CONSTRAINT `notas_ibfk_2` FOREIGN KEY (`grupoId`) REFERENCES `grupos`
(`grupoId`) ON DELETE CASCADE,
CONSTRAINT `invalidGradevalor` CHECK (`valor` >= 0 and `valor` <= 10),
CONSTRAINT `invalidconvocatoria` CHECK (`convocatoria` >= 1 and
`convocatoria` <= 3)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.notas: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `notas` DISABLE KEYS */;
/*!40000 ALTER TABLE `notas` ENABLE KEYS */;

-- Volcando estructura para tabla grados.profesordepartamento
DROP TABLE IF EXISTS `profesordepartamento`;
CREATE TABLE IF NOT EXISTS `profesordepartamento` (
`profesorDepartamentoId` int(11) NOT NULL AUTO_INCREMENT,
`profesorId` int(11) NOT NULL,
`departamentoId` int(11) NOT NULL,
PRIMARY KEY (`profesorDepartamentoId`),
KEY `profesorId` (`profesorId`),
KEY `departamentoId` (`departamentoId`),
CONSTRAINT `profesordepartamento_ibfk_1` FOREIGN KEY (`profesorId`)
REFERENCES `profesores` (`profesorId`),
CONSTRAINT `profesordepartamento_ibfk_2` FOREIGN KEY
(`departamentoId`) REFERENCES `departamentos` (`departamentoId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.profesordepartamento: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `profesordepartamento` DISABLE KEYS */;
```



```
/*!40000 ALTER TABLE `profesordepartamento` ENABLE KEYS */;

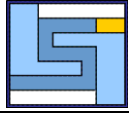
-- Volcando estructura para tabla grados.profesores
DROP TABLE IF EXISTS `profesores`;
CREATE TABLE IF NOT EXISTS `profesores` (
  `profesorId` int(11) NOT NULL AUTO_INCREMENT,
  `despachoId` int(11) NOT NULL,
  `departamentoId` int(11) NOT NULL,
  `dni` char(9) COLLATE latin1_spanish_ci NOT NULL,
  `nombre` varchar(100) COLLATE latin1_spanish_ci NOT NULL,
  `apellidos` varchar(100) COLLATE latin1_spanish_ci NOT NULL,
  `fechaNacimiento` date NOT NULL,
  `categoria` varchar(100) COLLATE latin1_spanish_ci NOT NULL,
  `emailProfesor` varchar(100) COLLATE latin1_spanish_ci NOT NULL,
  PRIMARY KEY (`profesorId`),
  UNIQUE KEY `dni` (`dni`),
  KEY `despachoId` (`despachoId`),
  KEY `departamentoId` (`departamentoId`),
  CONSTRAINT `profesores_ibfk_1` FOREIGN KEY (`despachoId`) REFERENCES
    `despachos` (`despachoId`),
  CONSTRAINT `profesores_ibfk_2` FOREIGN KEY (`departamentoId`)
    REFERENCES `departamentos` (`departamentoId`),
  CONSTRAINT `categoriaInvalida` CHECK (`categoria` in
    ('Catedratico','Titular de Universidad','Profesor Contrato
    Doctor','Profesor Ayudante Doctor'))
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.profesores: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `profesores` DISABLE KEYS */;
/*!40000 ALTER TABLE `profesores` ENABLE KEYS */;

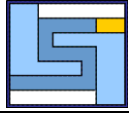
-- Volcando estructura para tabla grados.profesortutoria
DROP TABLE IF EXISTS `profesortutoria`;
CREATE TABLE IF NOT EXISTS `profesortutoria` (
  `profesorTutoriaId` int(11) NOT NULL AUTO_INCREMENT,
  `citaId` int(11) NOT NULL,
  `tutoriaId` int(11) NOT NULL,
  PRIMARY KEY (`profesorTutoriaId`),
  KEY `citaId` (`citaId`),
  KEY `tutoriaId` (`tutoriaId`),
  CONSTRAINT `profesortutoria_ibfk_1` FOREIGN KEY (`citaId`) REFERENCES
    `citas` (`citaId`),
  CONSTRAINT `profesortutoria_ibfk_2` FOREIGN KEY (`tutoriaId`)
    REFERENCES `tutorias` (`tutoriaId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.profesortutoria: ~0 rows
(aproximadamente)
/*!40000 ALTER TABLE `profesortutoria` DISABLE KEYS */;
/*!40000 ALTER TABLE `profesortutoria` ENABLE KEYS */;

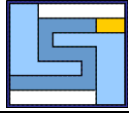
-- Volcando estructura para procedimiento grados.spInsertarAsignatura
DROP PROCEDURE IF EXISTS `spInsertarAsignatura`;
DELIMITER //
CREATE DEFINER=`iissi_user`@`localhost` PROCEDURE
`spInsertarAsignatura`(
  IN `curso` INT,
  IN `nombre` VARCHAR(255),
  IN `codigo` VARCHAR(7),
```

```
IN `creditos` INT,  
IN `tipo` VARCHAR(31),  
IN `departamento` VARCHAR(255),  
IN `vigencia` TINYINT  
  
)  
MODIFIES SQL DATA  
DETERMINISTIC  
BEGIN  
  
insert into asignaturas(  
curso,  
nombre,  
codigo,  
creditos,  
tipo,  
departamento,  
vigencia)  
values (  
curso,  
nombre,  
codigo,  
creditos,  
tipo,  
departamento,  
vigencia);  
END//  
DELIMITER ;  
  
-- Volcando estructura para procedimiento  
grados.spInsertarDatosAsignaturas  
DROP PROCEDURE IF EXISTS `spInsertarDatosAsignaturas`;  
DELIMITER //  
CREATE DEFINER=`iissi_user`@`localhost` PROCEDURE  
`spInsertarDatosAsignaturas`()  
MODIFIES SQL DATA  
BEGIN  
CALL spInsertarAsignatura('1', 'Álgebra Lineal y Numérica', '2060006',  
'6', 'Formación Básica', 'MATEMÁTICA APLICADA I', '1');  
CALL spInsertarAsignatura('1', 'Cálculo Infinitesimal y Numérico',  
'2060003', '6', 'Formación Básica', 'MATEMÁTICA APLICADA I', '1');  
CALL spInsertarAsignatura('1', 'Administración de Empresas',  
'2060002', '6', 'Formación Básica', 'ORGANIZACIÓN INDUSTRIAL Y GESTIÓN  
DE EMPRESAS I', '1');  
CALL spInsertarAsignatura('1', 'Circuitos Electrónicos Digitales',  
'2060004', '6', 'Formación Básica', 'TECNOLOGÍA ELECTRÓNICA', '1');  
CALL spInsertarAsignatura('1', 'Estadística', '2060007', '6',  
'Formación Básica', 'ESTADÍSTICA E INVESTIGACIÓN OPERATIVA', '1');  
CALL spInsertarAsignatura('1', 'Estructura de Computadores',  
'2060008', '6', 'Formación Básica', 'TECNOLOGÍA ELECTRÓNICA', '1');  
CALL spInsertarAsignatura('1', 'Fundamentos de Programación',  
'2060001', '12', 'Formación Básica', 'LENGUAJES Y SISTEMAS  
INFORMÁTICOS', '1');  
CALL spInsertarAsignatura('1', 'Fundamentos Físicos de la  
Informática', '2060009', '6', 'Formación Básica', 'FÍSICA APLICADA I',  
'1');  
CALL spInsertarAsignatura('1', 'Introducción a la Matemática  
Discreta', '2060005', '6', 'Formación Básica', 'MATEMÁTICA APLICADA  
I', '1');
```

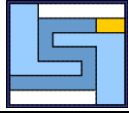
```
CALL spInsertarAsignatura('2', 'Análisis y Diseño de Datos y Algoritmos', '2060010', '12', 'Obligatoria', 'LENGUAJES Y SISTEMAS INFORMÁTICOS', '1');
CALL spInsertarAsignatura('2', 'Lógica Informática', '2060012', '6', 'Optativa', 'CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL', '1');
CALL spInsertarAsignatura('2', 'Matemática Discreta', '2060013', '6', 'Obligatoria', 'MATEMÁTICA APLICADA I', '1');
CALL spInsertarAsignatura('2', 'Redes de Computadores', '2060014', '6', 'Obligatoria', 'TECNOLOGÍA ELECTRÓNICA', '1');
CALL spInsertarAsignatura('2', 'Arquitectura de Computadores', '2060015', '6', 'Obligatoria', 'ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES', '1');
CALL spInsertarAsignatura('2', 'Arquitectura de Redes', '2060016', '6', 'Optativa', 'TECNOLOGÍA ELECTRÓNICA', '1');
CALL spInsertarAsignatura('2', 'Sistemas Operativos', '2060017', '6', 'Obligatoria', 'LENGUAJES Y SISTEMAS INFORMÁTICOS', '1');
CALL spInsertarAsignatura('2', 'Introducción a la Ingeniería del Software y los Sistemas de Información I', '2060054', '6', 'Obligatoria', 'LENGUAJES Y SISTEMAS INFORMÁTICOS', '1');
CALL spInsertarAsignatura('2', 'Introducción a la Ingeniería del Software y los Sistemas de Información II', '2060055', '6', 'Obligatoria', 'LENGUAJES Y SISTEMAS INFORMÁTICOS', '1');
CALL spInsertarAsignatura('3', 'Configuración Implementación y Mantenimiento de Sistemas Informáticos', '2060018', '6', 'Optativa', 'ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES', '1');
CALL spInsertarAsignatura('3', 'Gestión de Sistemas de Información', '2060019', '6', 'Optativa', 'LENGUAJES Y SISTEMAS INFORMÁTICOS', '1');
CALL spInsertarAsignatura('3', 'Gestión y Estrategia Empresarial', '2060020', '6', 'Optativa', 'ORGANIZACIÓN INDUSTRIAL Y GESTIÓN DE EMPRESAS I', '1');
CALL spInsertarAsignatura('3', 'Inteligencia Artificial', '2060021', '6', 'Obligatoria', 'CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL', '1');
CALL spInsertarAsignatura('3', 'Procesadores de Lenguajes', '2060022', '6', 'Optativa', 'LENGUAJES Y SISTEMAS INFORMÁTICOS', '1');
CALL spInsertarAsignatura('3', 'Programación Declarativa', '2060023', '6', 'Optativa', 'CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL', '1');
CALL spInsertarAsignatura('3', 'Tecnologías Avanzadas de la Información', '2060024', '6', 'Optativa', 'TECNOLOGÍA ELECTRÓNICA', '1');
CALL spInsertarAsignatura('3', 'Ampliación de Inteligencia Artificial', '2060025', '6', 'Optativa', 'CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL', '1');
CALL spInsertarAsignatura('3', 'Arquitectura de Sistemas Distribuidos', '2060026', '6', 'Optativa', 'ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES', '1');
CALL spInsertarAsignatura('3', 'Matemática Aplicada a Sistemas de Información', '2060027', '6', 'Optativa', 'MATEMÁTICA APLICADA I', '1');
CALL spInsertarAsignatura('3', 'Sistemas de Información Empresariales', '2060028', '6', 'Optativa', 'LENGUAJES Y SISTEMAS INFORMÁTICOS', '1');
CALL spInsertarAsignatura('3', 'Sistemas Inteligentes', '2060029', '6', 'Optativa', 'CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL', '1');
CALL spInsertarAsignatura('3', 'Sistemas Orientados a Servicios', '2060030', '6', 'Optativa', 'LENGUAJES Y SISTEMAS INFORMÁTICOS', '1');
CALL spInsertarAsignatura('4', 'Prácticas Externas', '2060031', '6', 'Optativa', 'LENGUAJES Y SISTEMAS INFORMÁTICOS', '1');
```



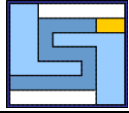
```
CALL spInsertarAsignatura('4', 'Acceso Inteligente a la Información',
'2060032', '6', 'Optativa', 'LENGUAJES Y SISTEMAS INFORMÁTICOS', '1');
CALL spInsertarAsignatura('4', 'Administración de Sistemas de
Información', '2060033', '6', 'Optativa', 'TECNOLOGÍA ELECTRÓNICA',
'1');
CALL spInsertarAsignatura('4', 'Gestión de Procesos y Servicios',
'2060034', '6', 'Optativa', 'LENGUAJES Y SISTEMAS INFORMÁTICOS', '1');
CALL spInsertarAsignatura('4', 'Infraestructura de Sistemas de
Información', '2060035', '6', 'Optativa', 'TECNOLOGÍA ELECTRÓNICA',
'1');
CALL spInsertarAsignatura('4', 'Integración de Sistemas Físicos e
Informáticos', '2060036', '6', 'Optativa', 'FÍSICA APLICADA I', '0');
CALL spInsertarAsignatura('4', 'Interacción Persona-ordenador',
'2060037', '6', 'Optativa', 'LENGUAJES Y SISTEMAS INFORMÁTICOS', '1');
CALL spInsertarAsignatura('4', 'Matemática Aplicada a Tecnologías de
la Información', '2060038', '6', 'Optativa', 'MATEMÁTICA APLICADA I',
'1');
CALL spInsertarAsignatura('4', 'Matemáticas para la Computación',
'2060039', '6', 'Optativa', 'MATEMÁTICA APLICADA I', '1');
CALL spInsertarAsignatura('4', 'Planificación y Gestión de Proyectos
Informáticos', '2060040', '6', 'Obligatoria', 'MATEMÁTICA APLICADA I',
'1');
CALL spInsertarAsignatura('4', 'Procesamiento de Imágenes Digitales',
'2060041', '6', 'Optativa', 'MATEMÁTICA APLICADA I', '1');
CALL spInsertarAsignatura('4', 'Seguridad en Sistemas Informáticos y
en Internet', '2060042', '6', 'Optativa', 'LENGUAJES Y SISTEMAS
INFORMÁTICOS', '1');
CALL spInsertarAsignatura('4', 'Teledetección', '2060043', '6',
'Optativa', 'TECNOLOGÍA ELECTRÓNICA', '1');
CALL spInsertarAsignatura('4', 'Aplicaciones de Soft Computing',
'2060044', '6', 'Optativa', 'ELECTRÓNICA Y ELECTROMAGNETISMO', '1');
CALL spInsertarAsignatura('4', 'Computación Móvil', '2060045', '6',
'Optativa', 'ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES', '1');
CALL spInsertarAsignatura('4', 'Criptografía', '2060046', '6',
'Optativa', 'MATEMÁTICA APLICADA I', '1');
CALL spInsertarAsignatura('4', 'Estadística Computacional', '2060047',
'6', 'Optativa', 'ESTADÍSTICA E INVESTIGACIÓN OPERATIVA', '1');
CALL spInsertarAsignatura('4', 'Gestión de la Producción', '2060048',
'6', 'Optativa', 'ORGANIZACIÓN INDUSTRIAL Y GESTIÓN DE EMPRESAS I',
'1');
CALL spInsertarAsignatura('4', 'Inteligencia Empresarial', '2060049',
'6', 'Optativa', 'LENGUAJES Y SISTEMAS INFORMÁTICOS', '1');
CALL spInsertarAsignatura('4', 'Modelado y Análisis de Requisitos en
Sistemas de Información', '2060050', '6', 'Optativa', 'LENGUAJES Y
SISTEMAS INFORMÁTICOS', '1');
CALL spInsertarAsignatura('4', 'Modelos de Computación y Complejidad',
'2060051', '6', 'Optativa', 'CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA
ARTIFICIAL', '1');
CALL spInsertarAsignatura('4', 'Tecnología Informática y Sociedad',
'2060052', '6', 'Optativa', 'TECNOLOGÍA ELECTRÓNICA', '1');
CALL spInsertarAsignatura('4', 'Trabajo Fin de Grado', '2060053',
'12', 'Trabajo fin de grado', 'TECNOLOGÍA ELECTRÓNICA', '1');

END//
DELIMITER ;

-- Volcando estructura para tabla grados.students
DROP TABLE IF EXISTS `students`;
CREATE TABLE IF NOT EXISTS `students` (
`studentId` int(11) NOT NULL AUTO_INCREMENT,
```



```
`accessMethod` varchar(30) COLLATE latin1_spanish_ci NOT NULL,  
`dni` char(9) COLLATE latin1_spanish_ci NOT NULL,  
`firstName` varchar(100) COLLATE latin1_spanish_ci NOT NULL,  
`surname` varchar(100) COLLATE latin1_spanish_ci NOT NULL,  
`birthDate` date NOT NULL,  
`email` varchar(250) COLLATE latin1_spanish_ci NOT NULL,  
PRIMARY KEY (`studentId`),  
UNIQUE KEY `dni` (`dni`),  
UNIQUE KEY `email` (`email`),  
CONSTRAINT `invalidStudentAccessMethod` CHECK (`accessMethod` in  
('Selectividad', 'Ciclo', 'Mayor', 'Titulado Extranjero'))  
) ENGINE=InnoDB AUTO_INCREMENT=22 DEFAULT CHARSET=latin1  
COLLATE=latin1_spanish_ci;  
  
-- Volcando datos para la tabla grados.students: ~21 rows  
(aproximadamente)  
/*!40000 ALTER TABLE `students` DISABLE KEYS */;  
REPLACE INTO `students` (`studentId`, `accessMethod`, `dni`,  
`firstName`, `surname`, `birthDate`, `email`) VALUES  
(1, 'Selectividad', '12345678A', 'Daniel', 'Pérez', '1991-01-01',  
'daniel@alum.us.es'),  
(2, 'Selectividad', '22345678A', 'Rafael', 'Ramírez', '1992-01-01',  
'rafael@alum.us.es'),  
(3, 'Selectividad', '32345678A', 'Gabriel', 'Hernández', '1993-01-01',  
'gabriel@alum.us.es'),  
(4, 'Selectividad', '42345678A', 'Manuel', 'Fernández', '1994-01-01',  
'manuel@alum.us.es'),  
(5, 'Selectividad', '52345678A', 'Joel', 'Gómez', '1995-01-01',  
'joel@alum.us.es'),  
(6, 'Selectividad', '62345678A', 'Abel', 'López', '1996-01-01',  
'abel@alum.us.es'),  
(7, 'Selectividad', '72345678A', 'Azael', 'González', '1997-01-01',  
'azael@alum.us.es'),  
(8, 'Selectividad', '8345678A', 'Uriel', 'Martínez', '1998-01-01',  
'uriel@alum.us.es'),  
(9, 'Selectividad', '92345678A', 'Gael', 'Sánchez', '1999-01-01',  
'gael@alum.us.es'),  
(10, 'Titulado Extranjero', '12345678B', 'Noel', 'Álvarez', '1991-02-02',  
'noel@alum.us.es'),  
(11, 'Titulado Extranjero', '22345678B', 'Ismael', 'Antúnez', '1992-02-02',  
'ismael@alum.us.es'),  
(12, 'Titulado Extranjero', '32345678B', 'Nathanael', 'Antolinez',  
'1993-02-02', 'nathanael@alum.us.es'),  
(13, 'Titulado Extranjero', '42345678B', 'Ezequiel', 'Aznárez', '1994-02-02',  
'ezequiel@alum.us.es'),  
(14, 'Titulado Extranjero', '52345678B', 'Ángel', 'Chávez', '1995-02-02',  
'angel@alum.us.es'),  
(15, 'Titulado Extranjero', '62345678B', 'Matusael', 'Gutiérrez',  
'1996-02-02', 'matusael@alum.us.es'),  
(16, 'Titulado Extranjero', '72345678B', 'Samael', 'Gálvez', '1997-02-02',  
'samael@alum.us.es'),  
(17, 'Titulado Extranjero', '82345678B', 'Baraquiél', 'Ibáñez', '1998-02-02',  
'baraquiél@alum.us.es'),  
(18, 'Titulado Extranjero', '92345678B', 'Otoniel', 'Idiáquez', '1999-02-02',  
'otoniel@alum.us.es'),  
(19, 'Titulado Extranjero', '12345678C', 'Niriel', 'Benítez', '1991-03-03',  
'niriel@alum.us.es'),  
(20, 'Titulado Extranjero', '22345678C', 'Múriel', 'Bermúdez', '1992-03-03',  
'muriel@alum.us.es'),
```

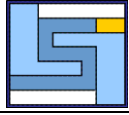


```
(21, 'Titulado Extranjero', '32345678C', 'John', 'AII', '2000-01-01',
'john@alum.us.es');
/*!40000 ALTER TABLE `students` ENABLE KEYS */;

-- Volcando estructura para tabla grados.subjects
DROP TABLE IF EXISTS `subjects`;
CREATE TABLE IF NOT EXISTS `subjects` (
  `subjectId` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(100) COLLATE latin1_spanish_ci NOT NULL,
  `acronym` varchar(8) COLLATE latin1_spanish_ci NOT NULL,
  `credits` int(11) NOT NULL,
  `course` int(11) NOT NULL,
  `type` varchar(20) COLLATE latin1_spanish_ci NOT NULL,
  `degreeId` int(11) NOT NULL,
  PRIMARY KEY (`subjectId`),
  UNIQUE KEY `name` (`name`),
  UNIQUE KEY `acronym` (`acronym`),
  KEY `degreeId` (`degreeId`),
  CONSTRAINT `subjects_ibfk_1` FOREIGN KEY (`degreeId`) REFERENCES
  `degrees` (`degreeId`),
  CONSTRAINT `negativeSubjectCredits` CHECK (`credits` > 0),
  CONSTRAINT `invalidSubjectCourse` CHECK (`course` > 0 and `course` <
6),
  CONSTRAINT `invalidSubjectType` CHECK (`type` in ('Formacion
Basica','Optativa','Obligatoria'))
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=latin1
COLLATE=latin1_spanish_ci;

-- Volcando datos para la tabla grados.subjects: ~13 rows
(aproximadamente)
/*!40000 ALTER TABLE `subjects` DISABLE KEYS */;
REPLACE INTO `subjects` (`subjectId`, `name`, `acronym`, `credits`,
`course`, `type`, `degreeId`) VALUES
(1, 'Diseño y Pruebas', 'DP', 12, 3, 'Obligatoria', 1),
(2, 'Acceso Inteligente a la Informacion', 'AII', 6, 4, 'Optativa',
1),
(3, 'Optimizacion de Sistemas', 'OS', 6, 4, 'Optativa', 1),
(4, 'Ingeniería de Requisitos', 'IR', 6, 2, 'Obligatoria', 1),
(5, 'Análisis y Diseño de Datos y Algoritmos', 'ADDA', 12, 2,
'Obligatoria', 1),
(6, 'Introducción a la Matematica Discreta', 'IMD', 6, 1, 'Formacion
Basica', 2),
(7, 'Redes de Computadores', 'RC', 6, 2, 'Obligatoria', 2),
(8, 'Teoría de Grafos', 'TG', 6, 3, 'Obligatoria', 2),
(9, 'Aplicaciones de Soft Computing', 'ASC', 6, 4, 'Optativa', 2),
(10, 'Fundamentos de Programación', 'FP', 12, 1, 'Formacion Basica',
3),
(11, 'Lógica Informatica', 'LI', 6, 2, 'Optativa', 3),
(12, 'Gestión y Estrategia Empresarial', 'GEE', 90, 3, 'Optativa', 3),
(13, 'Trabajo de Fin de Grado', 'TFG', 12, 4, 'Obligatoria', 3);
/*!40000 ALTER TABLE `subjects` ENABLE KEYS */;

-- Volcando estructura para tabla grados.tutorias
DROP TABLE IF EXISTS `tutorias`;
CREATE TABLE IF NOT EXISTS `tutorias` (
  `tutoriaId` int(11) NOT NULL AUTO_INCREMENT,
  `profesorId` int(11) NOT NULL,
  `dia` date NOT NULL,
  `horaComienzo` time NOT NULL,
  `horaFin` time NOT NULL,
```



```
PRIMARY KEY (`tutoriaId`),  
KEY `profesorId` (`profesorId`),  
CONSTRAINT `tutorias_ibfk_1` FOREIGN KEY (`profesorId`) REFERENCES  
`profesores` (`profesorId`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;  
  
-- Volcando datos para la tabla grados.tutorias: ~0 rows  
(aproximadamente)  
/*!40000 ALTER TABLE `tutorias` DISABLE KEYS */;  
/*!40000 ALTER TABLE `tutorias` ENABLE KEYS */;  
  
/*!40101 SET SQL_MODE=IFNULL(@OLD_SQL_MODE, '') */;  
/*!40014 SET FOREIGN_KEY_CHECKS=IF(@OLD_FOREIGN_KEY_CHECKS IS NULL, 1,  
@OLD_FOREIGN_KEY_CHECKS) */;  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```