

PRÁCTICA 1

Para la base de datos ejemplo:

1.- Listar los códigos de los directores de oficina.

SELECT DISTINCT dir

FROM oficinas

DIRECTOR

104

106

108

101

SELECT dir

FROM oficinas

DIRECTOR

104

106

108

106

104

101

106

104

108

** añadiendo la cláusula `DISTINCT' se evita que los códigos aparezcan repetidos.*

2. Listar toda la información de los empleados de la oficina 21.

*SELECT **

FROM oficinas

WHERE oficina = 21

OFICINA CIUDAD REGION DIR OBJETIVO VENTAS

21 Los Angeles Oeste 108 \$725,915.00 \$835,915.00

3. Listar nombre y fecha de contrato de los empleados que no están asignados a ninguna oficina.

SELECT nombre, contrato

FROM repventas

WHERE oficina_rep IS NULL

NOMBRE CONTRATO

Tom Snyder 13-ENE-90

4. Listar todos los pedidos del cliente 2114 tomados por empleado 102 ordenados por importe y fecha

*SELECT **

FROM pedidos

WHERE clie = 2114 AND

rep = 102

ORDER BY importe, fecha

NUM_PEDIDO FECHA_PEDIDO CLIE REP FAB PRODUCTO CANT IMPORTE

112979 12-OCT-89 2114 102 ACI 4100Z 6 \$15,000.00

5. Listado de los vendedores asignados a la oficina 11.

```

SELECT nombre
FROM repventas
WHERE oficina = 11

NOMBRE

```

Mary Jones
 Sam Clark

6.Listado de los vendedores que tomaron pedidos el día 2 de febrero de 1990.

```

SELECT rep
FROM pedidos
WHERE fecha_pedido = 02-feb-90

REP

```

108
 101

7.Listar las empresas con pedidos superiores a 2.000\$ y su límite de crédito.

```

SELECT empresa, limite_credito
FROM clientes, pedidos
WHERE clie = num_clie AND
importe >= 2000

```

EMPRESA LIMITE_CREDITO

JCP Inc. \$50,000.00
 Firts Corp \$65,000.00
 Acme Mfg. \$50,000.00
 Ace Intenational. \$35,000.00

Zetacorp. \$50,000.00

Orion Corp. \$20,000.00

Peter Brothers \$40,000.00

Holm & Landis \$55,000.00

J.P. Sinclair \$35,000.00

Rico Enterprises \$50,000.00

Fred Lewis Corp \$65,000.00

Ian & Schmidt \$20,000.00

Chen Associates \$25,000.00

8. Listar el nombre de cada empleado y el de su director.

SELECT nombre, director

FROM repventas

NOMBRE DIRECTOR

Bill Adams 104

Mary Jones 106

Sue Smith 108

Sam Clark NULL

Bob Smith 106

Dan Roberts 104

Tom Snyder 101

Larry Fitch 106

Paul Cruz 104

Nancy Angelli 108

9. Listar los pedidos superiores a 500\$ realizados al vendedor Paul Cruz indicANDO el cliente que realizó el pedido.

SELECT num_pedido, clie

FROM pedidos

WHERE importe >= 500 AND

rep=103

NUM_PEDIDO CLIE

112975 2111

113057 2111

10.INSERTar una nueva oficina en Boston, de número 24, cuyo director sea el empleado 106, con unos objetivos de ventas de 350.000\$.

INSERT INTO oficinas(ciudad, oficina, dir, objetivo)

VALUES('Boston',24,106,350.000)

11.INSERTar como nuevo representante de ventas a Tom Smith (código de empleado 111), de 35 años, asignado a la oficina de Boston, con una cuota de ventas de 100.000\$ y con título de representante de ventas.

INSERT INTO repventas(nombre, num_empleado, edad, cuota, titulo, oficina_rep)

VALUES ('Tom Smith', 111, 35, 100000, 'repventas', 24)

12.Asignar unas ventas de 100.000\$ a la oficina de Boston y modificar su objetivo de ventas, situándolo en 500.000\$.

UPDATE oficinas

SET ventas = 100000 AND

objetivo = 500000

WHERE ciudad = 'Boston'

13. Insertar varios pedidos de importe 100\$. Comprobar la operación. Borrar esos pedidos.

INSERT INTO pedidos(num_pedido, fecha_pedido, clie, rep, fab, producto, can, importe)

VALUES (113043, `26-ene-99`, 2113, 101, QSA, XK47, 20, \$100)

INSERT INTO pedidos(num_pedido, fecha_pedido, clie, rep, fab, producto, can, importe)

VALUES (113044, `26-ene-99`, 2113, 101, QSA, XK48, 20, \$100)

INSERT INTO pedidos(num_pedido, fecha_pedido, clie, rep, fab, producto, can, importe)

VALUES (113045, '26-ene-99', 2113, 101, QSA, XK48A, 20, \$100)

SELECT *

FROM pedidos

WHERE importe = \$100

NUM_PEDIDO FECHA_PEDIDO CLIE REP FAB PRODUCTO CANT IMPORTE

113043 26-ENE-99 2113 101 QSA XK47 20 \$100.00

113044 26-ENE-99 2113 101 QSA XK48 20 \$100.00

113045 26-ENE-99 2113 101 QSA XK48A 20 \$100.00

DELETE

FROM pedidos

WHERE importe = \$100

14.Borrar el empleado Tom Smith (código 111), y eliminar la oficina de Boston.

DELETE

FROM repventas

WHERE num_empl = 111

DELETE

FROM oficinas

WHERE ciudad = 'Boston'

15.Comprobar que la base de datos ha quedado como estaba al iniciar la práctica.

Hemos comprobado, con un 'SELECT *' de cada una de las tablas que su contenido es el mismo descrito en el libro.

PRÁCTICA 2

Diseñar e implementar una sencilla base de datos que permita almacenar información sobre alumnos de la Escuela y las asignaturas cursadas por estos. Se debe poder consultar el expediente académico de un alumno/a (asignaturas cursadas, calificación, convocatoria y año en que se aprobó). Para ello se crearán las tablas correspondientes, y se diseñarán varias consultas, como pueden ser

Calificaciones de un/a alumno/a en un determinado curso

Alumnos/as matriculados en una asignatura

Calificaciones de todos/as los/as alumnos/as matriculados en determinado curso académico (por ejemplo curso 96/97).

Para comprobar la viabilidad el diseño realizado, insertar varios datos en las tablas y realizar las consultas anteriores.

Código SQL para la creación de las tablas

Tabla ig_asignaturas

```
CREATE TABLE ig_asignaturas  
(datos_asignatura VARCHAR(15) NOT NULL,  
cod_asignatura INTEGER NOT NULL,  
PRIMARY KEY (cod_asignatura))
```

Tabla ig_alumnos

```
CREATE TABLE ig_alumnos  
(datos_alumno VARCHAR(15) NOT NULL,  
cod_alumno INTEGER NOT NULL,  
PRIMARY KEY (cod_alumno))
```

Tabla ig_expedientes

```
CREATE TABLE ig_expediente  
(cod_expediente INTEGER NOT NULL,  
alumno INTEGER NOT NULL,  
asignatura INTEGER NOT NULL,  
nota float NOT NULL,  
convocatoria VARCHAR(15) NOT NULL,  
año INTEGER NOT NULL,  
PRIMARY KEY (cod_expediente),  
FOREIGN KEY (alumno)  
references alumnos.cod_alumno
```

ON DELETE RESTRICT,

FOREIGN KEY (asignatura)

references asignaturas.cod_asignatura

ON DELETE RESTRICT)

Para comprobar la viabilidad el diseño realizado, insertar varios datos en las tablas y realizar las consultas anteriores.

Tabla ig_asignaturas

INSERT INTO ig_asignaturas(datos_asignatura, cod_asignatura)

VALUES('matemáticas', 01)

INSERT INTO ig_asignaturas(datos_asignatura, cod_asignatura)

VALUES('lengua', 02)

INSERT INTO ig_asignaturas(datos_asignatura, cod_asignatura)

VALUES('fisica', 03)

INSERT INTO ig_asignaturas(datos_asignatura, cod_asignatura)

VALUES('quimica', 04)

*SELECT **

FROM ig_asignaturas

datos_asignatura cod_asignatura

matematicas 1

lengua 2

fisica 3

quimica 4

Tabla ig_alumnos.

INSERT INTO ig_alumnos(datos_alumno, cod_alumno)

VALUES('Guillermo de Haro', 01)

INSERT INTO ig_alumnos(datos_alumno, cod_alumno)

VALUES('Ines Vidal', 02)

INSERT INTO ig_alumnos(datos_alumno, cod_alumno)

VALUES('Cris Viñuela', 03)

INSERT INTO ig_alumnos(datos_alumno, cod_alumno)

VALUES('Sergio de Miguel', 04)

SELECT *

FROM ig_alumnos

datos_alumno cod_alumno

Guillermo de Ha 1

Ines Vidal 2

Cris Viñuela 3

Sergio de Migue 4

Tabla ig_expediente.

INSERT INTO ig_expediente(nota, convocatoria, año, cod_expediente, alumno, asignatura)

VALUES(7, 'Junio', 98, 01, 4, 1)

INSERT INTO ig_expediente(nota, convocatoria, año, cod_expediente, alumno, asignatura)

VALUES(5, 'Febrero', 99, 02, 4, 3)

INSERT INTO ig_expediente(nota, convocatoria, año, cod_expediente, alumno, asignatura)

VALUES(6, 'Septiembre', 99, 04, 4, 4)

INSERT INTO ig_expediente(nota, convocatoria, año, cod_expediente, alumno, asignatura)

VALUES(7, 'Junio', 98, 05, 3, 2)

INSERT INTO ig_expediente(nota, convocatoria, año, cod_expediente, alumno, asignatura)

VALUES(6, 'Febrero', 99, 06, 3, 3)

INSERT INTO ig_expediente(nota, convocatoria, año, cod_expediente, alumno, asignatura)

VALUES(5, 'Septiembre', 99, 07, 3, 4)

INSERT INTO ig_expediente(nota, convocatoria, año, cod_expediente, alumno, asignatura)

VALUES(7, 'Junio', 98, 08, 2, 1)

INSERT INTO ig_expediente(nota, convocatoria, año, cod_expediente, alumno, asignatura)

VALUES(6, 'Febrero', 99, 09, 2, 2)

INSERT INTO ig_expediente(nota, convocatoria, año, cod_expediente, alumno, asignatura)

VALUES(5, 'Septiembre', 99, 10, 2, 3)

INSERT INTO ig_expediente(nota, convocatoria, año, cod_expediente, alumno, asignatura)

VALUES(7, 'Junio', 98, 11, 1, 3)

INSERT INTO ig_expediente(nota, convocatoria, año, cod_expediente, alumno, asignatura)

VALUES(6, 'Febrero', 99, 12, 1, 1)

INSERT INTO ig_expediente(nota, convocatoria, año, cod_expediente, alumno, asignatura)

VALUES(5, 'Septiembre', 99, 13, 1, 4)

*SELECT **

FROM ig_expediente

nota convocatoria año cod_expediente alumno asignatura

7.00 Junio 98 1 4 1

7.00 Febrero 99 2 3 3

6.00 Septiembre 99 4 4 4

7.00 Junio 98 5 3 2

6.00 Febrero 99 6 3 3

5.00 Septiembre 99 7 3 4

7.00 Junio 98 8 2 1

6.00 Febrero 99 9 2 2

5.00 Septiembre 99 10 2 3

7.00 Junio 98 11 1 3

6.00 Febrero 99 12 1 1

5.00 Septiembre 99 13 1 4

Realizar las siguientes consultas:

Calificaciones de un/a alumno/a en un determinado curso

SELECT datos_asignatura, datos_alumno, nota, año

FROM ig_asignaturas, ig_alumnos, ig_expediente

WHERE cod_alumno = alumno AND

cod_asignatura = asignatura AND

datos_alumno = 'Cris Viñuela' AND

año = 98

datos_asignatura datos_alumno nota año

lengua Cris Viñuela 7.00 98

Alumnos/as matriculados en una asignatura

SELECT datos_asignatura, datos_alumno

FROM ig_asignaturas, ig_alumnos, ig_expediente

WHERE cod_alumno = alumno AND

cod_asignatura = asignatura

datos_asignatura datos_alumno

fisica Guillermo de Ha

matematicas Guillermo de Ha

quimica Guillermo de Ha

matematicas Ines Vidal

lengua Ines Vidal

fisica Ines Vidal

fisica Cris Viñuela

lengua Cris Viñuela

fisica Cris Viñuela

quimica Cris Viñuela

matematicas Sergio de Migue

quimica Sergio de Migue

Calificaciones de todos/as los/as alumnos/as matriculados en determinado curso académico (por ejemplo curso 96/97).

```
SELECT datos_asignatura, datos_alumno, año
```

```
FROM ig_asignaturas, ig_alumnos, ig_expediente
```

```
WHERE cod_alumno = alumno AND
```

```
cod_asignatura = asignatura AND
```

```
año = 98
```

```
datos_asignatura datos_alumno año
```

fisica Guillermo de Ha 98

matematicas Ines Vidal 98

lengua Cris Viñuela 98

matematicas Sergio de Migue 98

PRACTICA 3

Se quiere utilizar una base de datos para llevar la gestión de un hospital. Debe permitir conocer la situación de las camas disponibles, los enfermos actualmente en el hospital, el pabellón en el que se encuentran, su historial médico, los médicos que trabajan en él, sus especialidades, etc.

La base de datos debe permitir realizar las consultas que se requieren habitualmente, como pueden ser:

Camas libres en un pabellón.

Historial médico de un paciente, incluyendo estancias anteriores y médicos que le trataron.

Localización de un paciente en el hospital.

Listado de médicos por especialidades.

Lista de pacientes actualmente ingresados atendidos por un determinado médico.

Así como insertar los datos correspondientes a un paciente que ingresa en el hospital.

Se pide:

a) Realizar el diagrama entidad/relación.

Las relaciones arriba expuestas se refieren a una fecha concreta. En ese caso se considera que:

Un médico puede atender a varios pacientes a la vez, al igual que varios pueden atender a un solo paciente.

Un enfermo puede ocupar una y solo una cama en cada una de las estancias en el hospital.

b) Paso del diagrama entidad/relación al modelo relacional. Definición de las tablas.

•
c) Normalización de las tablas (hasta 3NF)

1FN:

No hay grupos repetitivos.

2FN:

Todos los atributos dependen de la clave primaria.

3FN:

Todos los atributos dependen directamente de la clave primaria.

- **Escribir las sentencias SQL que permiten crear las tablas, así como realizar las consultas y actualizaciones descritas anteriormente.**

Tabla ig_enfermos.

```
CREATE TABLE ig_enfermos  
(cod_enfermo INTEGER NOT NULL,  
datos_enfermo VARCHAR(15) NOT NULL,  
PRIMARY KEY (cod_enfermo))
```

Tabla ig_camias.

```
CREATE TABLE ig_camias  
(cod_cama INTEGER NOT NULL,  
pabellon VARCHAR(5) NOT NULL,
```

PRIMARY KEY (cod_cama))

Tabla ig_médicos.

CREATE TABLE i g_medicos

*(cod_medico INTEGER NOT NULL,
datos_medico VARCHAR(15) NOT NULL,
especialidad VARCHAR(15) NOT NULL,
PRIMARY KEY (cod_medico))*

Tabla ig_historiales.

CREATE TABLE ig_historiales

*(enfermo INTEGER NOT NULL,
fecha DATE NOT NULL,
estancia_dias INTEGER NOT NULL,
patología VARCHAR(30) NOT NULL,
medico INTEGER NOT NULL,
cama INTEGER NOT NULL,
PRIMARY KEY (enfermo, fecha, medico),
FOREIGN KEY (enfermo)
references cod_enfermo
ON DELETE RESTRICT,
FOREIGN KEY (medico)
references cod_medico
ON DELETE RESTRICT,
FOREIGN KEY (cama)
references cod_cama
ON DELETE RESTRICT)*

Tabla ig_camás.

```
CREATE TABLE ig_camas
```

```
(cod_cama INTEGER NOT NULL,
```

```
pabellon VARCHAR(5) NOT NULL,
```

```
PRIMARY KEY (cod_cama))
```

*** Un enfermo puede ocupar sólo una cama, pero dado que si establecemos la restricción de unicidad debemos imponer también que el campo no sea nulo y las camas no tienen por que estar siempre ocupadas. Por tanto sería necesaria comprobación adicional para asegurar que no se producen errores de asignación en este sentido.*

Tabla ig_enfermos.

```
INSERT INTO ig_enfermos (cod_enfermo, datos_enfermo)
```

```
VALUES (1,'José Torrado')
```

```
INSERT INTO ig_enfermos (cod_enfermo, datos_enfermo)
```

```
VALUES (2,'Juan García')
```

```
INSERT INTO ig_enfermos (cod_enfermo, datos_enfermo)
```

```
VALUES (3,'Amadeo Torres')
```

```
INSERT INTO ig_enfermos (cod_enfermo, datos_enfermo)
```

```
VALUES (4,'Vicente Vega')
```

```
SELECT * FROM ig_enfermos
```

```
cod_enfermo datos_enfermo
```

```
-----
```

```
1 José Torrado
```

```
2 Juan García
```

```
3 Amadeo Torres
```

```
4 Vicente Vega
```

Tabla ig_camas

```
INSERT INTO ig_camas(cod_cama, pavellon)
```

```
VALUES(1, 'A')
```

```
INSERT INTO ig_camas(cod_cama, pavellon)
```

VALUES(2, 'B')

INSERT INTO ig_camas(cod_cama, pavellon)

VALUES(3, 'C')

INSERT INTO ig_camas(cod_cama, pavellon)

VALUES(4, 'A')

INSERT INTO ig_camas(cod_cama, pavellon)

VALUES(5, 'D')

INSERT INTO ig_camas(cod_cama, pavellon)

VALUES(6, 'C')

SELECT * FROM ig_camas

cod_cama pavellon

1 A

2 B

3 C

4 A

5 D

6 C

Tabla médicos.

INSERT INTO ig_medicos(cod_medico, datos_medico, especialidad)

VALUES(1, 'Bernardo López', 'Cirugía')

INSERT INTO ig_medicos(cod_medico, datos_medico, especialidad)

VALUES(2, 'Andrés Calamaro', 'Respiratorio')

INSERT INTO ig_medicos(cod_medico, datos_medico, especialidad)

VALUES(3, 'Manuel García', 'Circulatorio')

INSERT INTO ig_medicos(cod_medico, datos_medico, especialidad)

VALUES(4,'Ramón Rodríguez', 'Traumatología')

INSERT INTO ig_medicos(cod_medico, datos_medico, especialidad)

VALUES(5,'Pepe Pérez', 'Traumatología')

SELECT * FROM ig_medicos

cod_medico datos_medico especialidad

1 Bernardo López Cirugía

2 ANDrés Calamaro Respiratorio

3 Manuel García Circulatorio

4 Ramón Rodríguez Traumatología

5 Pepe Pérez Traumatología

Tabla ig_historiales.

INSERT INTO ig_historiales (enfermo, fecha, estancia_dias, patología, medico, cama)

VALUES (1, '1995-01-25',15, 'Traumatismo craneal', 4, 5)

INSERT INTO ig_historiales (enfermo, fecha, estancia_dias, patología, medico, cama)

VALUES (1, '1995-01-25',15, 'Traumatismo craneal', 5, 5)

INSERT INTO ig_historiales (enfermo, fecha, estancia_dias, patología, medico, cama)

VALUES (1, '1997-01-25', 2, 'Ulcera', 1, 2)

INSERT INTO ig_historiales (enfermo, fecha, estancia_dias, patología, medico, cama)

VALUES (1, '1997-01-25', 2, 'Ulcera', 3, 2)

INSERT INTO ig_historiales (enfermo, fecha, estancia_dias, patología, medico, cama)

VALUES (1, '1999-01-25', 3, 'Operación cornea', 1, 3)

INSERT INTO ig_historiales (enfermo, fecha, estancia_dias, patología, medico, cama)

VALUES (2, '1998-02-30', 14, 'Apendicitis', 1, 6)

INSERT INTO ig_historiales (enfermo, fecha, estancia_dias, patología, medico, cama)

VALUES (2, '1999-02-09', 14, 'Laringitis', 2, 5)

```

INSERT INTO ig_historiales (enfermo, fecha, estancia_dias, patología, medico, cama)
VALUES (3, '1996-07-28', 29, 'Pulmonía', 2, 2)

INSERT INTO ig_historiales (enfermo, fecha, estancia_dias, patología, medico, cama)
VALUES (3, '2000-01-15', 19, 'Fractura de Pelvis', 1, 4)

INSERT INTO ig_historiales (enfermo, fecha, estancia_dias, patología, medico, cama)
VALUES (3, '2000-01-15', 19, 'Fractura de Pelvis', 4, 1)

INSERT INTO ig_historiales (enfermo, fecha, estancia_dias, patología, medico, cama)
VALUES (3, '2000-01-15', 19, 'Fractura de Pelvis', 5, 1)

INSERT INTO ig_historiales (enfermo, fecha, estancia_dias, patología, medico, cama)
VALUES (4, '1998-08-12', 45, 'Leucemia', 3, 4)

INSERT INTO ig_historiales (enfermo, fecha, estancia_dias, patología, medico, cama)
VALUES (4, '1998-08-12', 45, 'Leucemia', 1, 4)

INSERT INTO ig_historiales (enfermo, fecha, estancia_dias, patología, medico, cama)
VALUES (4, '1998-09-12', 45, 'Arterioesclerosis', 3, 2)

SELECT * FROM ig_historiales

enfermo fecha estancia_dias patología medico cama

```

1 1995-01-25 15 Traumatismo craneal 4 5

1 1995-01-25 15 Traumatismo craneal 5 5

1 1997-01-25 2 Ulcera 1 2

1 1997-01-25 2 Ulcera 3 2

1 1999-01-25 3 Operación cornea 1 3

2 1998-02-30 14 Apendicitis 1 6

2 1999-02-09 14 Laringitis 2 5

3 1996-07-28 29 Pulmonía 2 2

3 2000-01-15 19 Fractura de Pelvis 1 4

3 2000-01-15 19 Fractura de Pelvis 4 1

3 2000-01-15 19 Fractura de Pelvis 5 1

4 1998-08-12 45 Leucemia 3 4

4 1998-08-12 45 Leucemia 1 4

4 1998-09-12 45 Arterioesclerosis 3 2

Camas libres en un pabellón.

```
SELECT cod_cama, pavellon
```

```
FROM ig_camas
```

```
WHERE cod_cama NOT IN
```

```
(SELECT cod_cama, pavellon
```

```
FROM ig_camas, ig_historial
```

```
WHERE cod_cama = cama AND
```

```
AND ((to_days('2000-01-26') - estancia_dias) <= to_days(fecha)))
```

* No se ha podido comprobar el funcionamiento porque el servidor de Bases de Datos con el que se realizan las prácticas no acepta subconsultas.

Historial médico de un paciente, incluyendo estancias anteriores y médicos que le trataron.

```
SELECT datos_enfermo, fecha, estancia_dias, patología, datos_medico
```

```
FROM ig_enfermos, ig_medicos, ig_historiales
```

```
WHERE cod_enfermo = enfermo AND
```

```
cod_medico = medico AND
```

```
datos_enfermo = 'José Torrado'
```

```
datos_enfermo fecha estancia_dias patología datos_medico
```

José Torrado 1997-01-25 2 Ulcera Bernardo López

José Torrado 1999-01-25 3 Operación cornea Bernardo López

José Torrado 1997-01-25 2 Ulcera Manuel García

José Torrado 1995-01-25 15 Traumatismo craneal Ramón Rodríguez

Localización de un paciente en el hospital.

SELECT datos_enfermo, pavellon, cod_cama

FROM ig_enfermos, ig_camas, ig_historiales

WHERE cod_enfermo = enfermo AND

cod_cama= cama AND

datos_enfermo = 'José Torrado'

datos_enfermo pavellon cod_cama

José Torrado B 2

José Torrado B 2

José Torrado C 3

José Torrado D 5

José Torrado D 5

Listado de médicos por especialidades.

SELECT datos_medico, especialidad

FROM ig_medicos

ORDER BY especialidad

datos_medico especialidad

Manuel García Circulatorio

Bernardo López Cirugía

ANDrés Calamaro Respiratorio

Ramón Rodríguez Traumatología

Pepe Pérez Traumatología

Lista de pacientes actualmente ingresados atendidos por un determinado médico.

```

SELECT datos_medico, datos_enfermo, fecha, estancia_dias
FROM ig_medicos, ig_enfermos, ig_historiales
WHERE cod_medico = medico AND
cod_enfermo = enfermo AND
((to_days('2000-01-26') - estancia_dias) <= to_days(fecha))
datos_medico datos_enfermo fecha estancia_dias

```

Bernardo López Amadeo Torres 2000-01-15 19

Ramón Rodríguez Amadeo Torres 2000-01-15 19

Pepe Pérez Amadeo Torres 2000-01-15 19

ALUMNOS

ASIGNATURAS

M:N

Estudian

Estudiadas por

Modelo Entidad/Relación.

ASIGNATURAS
cod_asignatura #
datos_asignatura

EXPEDIENTE
cod_expediente #
alumno
asignatura
nota
convocatoria
año

ALUMNOS
cod_alumno #
datos_alumno

Modelo Relacional

Modelo Relacional

CAMAS
cod_cama #
pabellon

ENFERMOS
cod_enfermo #
datos_enfermo

HISTORIALES
enfermo #
medico #
fecha #
estancia_dias
patología
cama

Modelo Entidad/Relación.

Ocupada por

Atiende

M:N

CAMAS

MÉDICOS

ENFERMOS

1:1

Ocupa

Atendido por

MEDICOS
cod_medico #
datos_medico
especialidad