



CLASES PRÁCTICAS BD *(2º GIIGSI)*

Conchi Presedo García
E.Ingeniería de Bilbao – 2021/2022

INDICE

• ENUNCIADOS PRÁCTICAS:

MS ACCESS (de P1 a P4)

P1: Inicio con Access, creación de una BD, tablas, claves y relaciones	5
P2: Consultas sencillas en vista Diseño, formularios e informes	9
P3: Más consultas, importación de datos e integridad referencial	17
P4: Creación y carga de datos en una BD con integridad referencial	25

MySQL (de P5 a P7)

P5: Inicio con XAMPP y MySQL	33
P6: LDD y actualización de datos en SQL	41
P7: Consultas en SQL y funciones de fecha y hora con MySQL	43

PROYECTO VIDEOCLUB (de P8 a P12)

P8: Enunciado y tareas del proyecto	49
P9: Aplicación Web ASP.NET con VB.NET y MS ACCESS	55
P10: Controles de servidor ASP.NET para las páginas Web	59
P11: Consulta SELECT en MS ACCESS y en la aplicación	65
P12: Programación de las páginas Web del socio y del administrador	69

• ANEXOS:

– BD UNIVERSIDAD	75
– BD REPRES_VENTAS	77
– BD EMPRESA	81
– BD BUQUES	83
– BD BIBLIOTECA	85

PRÁCTICA P1

MS ACCESS (1):

Inicio con Access, creación de una BD, tablas, claves y relaciones

OBJETIVOS

Para realizar esta práctica se necesita:

- No hay actividades previas.

Al terminar esta práctica se debe saber:

- Cómo entrar en Access.
- Cómo crear y guardar una base de datos.
- Qué es una tabla, cómo crearla sin asistente y cómo introducir datos en ella.
- Qué es una clave primaria y cómo crearla.
- Qué es una relación o vínculo entre tablas y cómo crearlo.

NOTA: Existe ayuda sobre las tareas a realizar en esta práctica al final de la práctica P4.
--

CREAR UNA BASE DE DATOS (BD) EN ACCESS

- Crear una base de datos nueva que se llame UNI_<primer apellido>. Ej: UNI_PRESEDO.

CREAR LAS TABLAS DE LA BD

- Crear las tablas de la BD UNIVERSIDAD (Página 6 del Tema 1 de Teoría):

ALUMNO, con los campos:

- Nombre : Texto 20 caracteres máximo. Apellido del alumno.
- Código alumno: Número que identifica al alumno.
- Año: Curso en el que está. Comprobar que sea 1, 2, 3 ó 4. Por defecto: 1.
- Especialidad: Código de los estudios, CS → Informática, IT→ Ingeniería Técnica Industrial.
Comprobar que sea CS ó IT.

CURSO, con los campos:

- Nombre curso: Texto de 30 caracteres máximo con el nombre de la asignatura.
- Código curso: Código alfanumérico que identifica la asignatura.
- Créditos: Correspondientes a esa asignatura.
- Departamento: Dpto responsable de esa asignatura.

SECCIÓN, con los campos:

- Identificador sección: Código numérico que identifica la sección.
- Código curso: Código alfanumérico que identifica la asignatura.
- Semestre: En el que se imparte la asignatura. Comprobar que sea primavera u otoño.
- Año: En el que se ha impartido esa sección. Por defecto: Año actual.
- Profesor: Texto 20 caracteres máximo. Apellido del profesor.

INFORME_CALIFICACIONES, con los campos:

- Código alumno: Número que identifica al alumno.
- Identificador sección: Código numérico que identifica la sección.
- Calificación: Obtenida por el alumno en esa sección. Comprobar que sea: A, B, C ó D.

REQUISITO, con los campos:

- Código curso: Código alfanumérico que identifica la asignatura.
- Código requisito: Código de la asignatura que es requisito de la anterior.

MATRICULA, con los campos:

- Código alumno: Número que identifica al alumno.
- Precio matrícula: Coste total de la matrícula de ese alumno.
- Fecha_Hora: Fecha y hora de la realización de esa matrícula. Por defecto: Fecha y hora actual.
- Observaciones: Texto (largo) con las incidencias en la matrícula de ese alumno.

INTRODUCIR LOS SIGUIENTES DATOS EN LAS TABLAS

ALUMNO

Nombre	Código alumno	Año	Especialidad
Smith	17	1	CS
Brown	8	2	CS
Juan	22	1	CS
Pepe	23	3	IT

INFORME_CALIFICACIONES

Código alumno	Identificador sección	Calificación
8	85	A
8	92	A
8	102	B
8	135	A
17	112	B
17	119	C
22	112	A
22	135	A
23	112	D

CURSO

Nombre Curso	Código Curso	Créditos	Departamento
Introd. a la computación	CS1310	4	LSI
Estructuras de datos	CS3320	4	LSI
Matemáticas Discretas	MATE2410	3	MATE
Bases de Datos	CS3380	6	LSI
Administración de BD	CS3390	6	LSI
Diseño de BD	CS3400	6	LSI

REQUISITO

Código curso	Código requisito
CS3380	CS3320
CS3380	MATE2410
CS3320	CS1310
CS3390	CS3380
CS3400	CS3380

SECCIÓN

Identificador sección	Código Curso	Semestre	Año	Profesor
85	MATE2410	Otoño	1998	King
92	CS1310	Otoño	1998	Anderson
102	CS3320	Primavera	1999	Knuth
112	MATE2410	Otoño	1999	Chang
119	CS1310	Otoño	1999	Anderson
135	CS3380	Otoño	1999	Stone

MATRICULA

Código alumno	Precio	Fecha_Hora	Observaciones
8	1000	1/10/2002 10:00:00	Falta fotocopia del carné de identidad.
17	800	15/9/2002 12:15:00	

ESTABLECER LAS CLAVES PRIMARIAS DE LA BD.

ESTABLECER LAS RELACIONES OPORTUNAS ENTRE LAS TABLAS ANTERIORES

PRÁCTICA P2

MS ACCESS (2):

Consultas sencillas en vista Diseño, formularios e informes

OBJETIVOS

Para realizar esta práctica se necesita:

- La BD UNIVERSIDAD creada en la práctica 1.

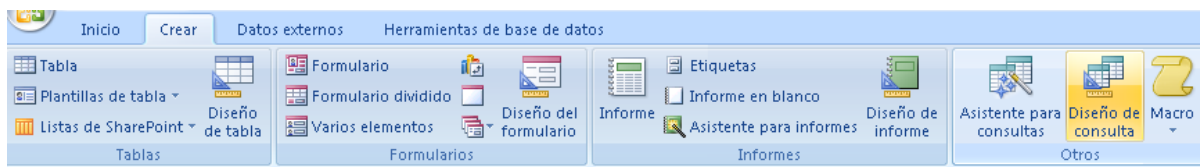
Al terminar esta práctica se debe saber:

- Realizar consultas sencillas con la vista Diseño de Access
- Crear y manejar subformularios en Access.
- Crear y manejar informes y subinformes.
- Nociones básicas sobre Macros.
- Diseño de páginas web en Html con asistente.

CONSULTAS SENCILLAS

PASOS a dar para hacer consultas a una base de datos en MS ACCESS:

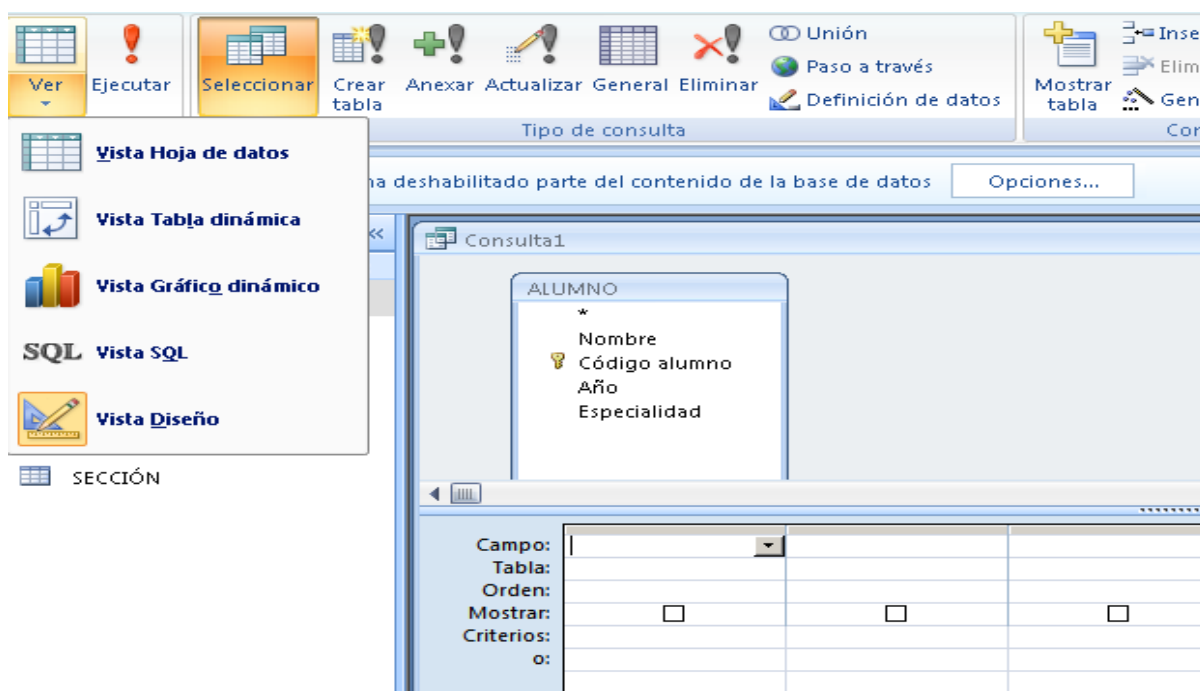
Paso 1. Crear una consulta nueva con Pestaña *Crear* → *Diseño de Consulta*



Paso 2. Añadir la tabla o tablas que se necesiten en la consulta

Paso 3. Especificar la consulta empleando la *Vista Diseño* (lenguaje llamado QBE ó Query By Example) o bien la *Vista SQL* (lenguaje llamado SQL ó Standard Query Language, que es el estándar de las bases de datos)

Paso 4. Ejecutar la consulta pulsando el icono *Ejecutar*. Los resultados se verán en la Vista Hoja de Datos de la consulta



Realizar las siguientes **CONSULTAS** sobre la tabla ALUMNO en la que habremos añadido al menos dos registros nuevos (Juan, 12, 3, IT) y (Pepe, 21, 1, CS):

- Ejemplo para usar la vista SQL: Recuperar las especialidades cursadas por los alumnos. (debe dar CS e IT una sola vez)
- Ejemplo con una sola condición: Recuperar el/los alumno/s cuyo apellido es 'Smith'.
- Ejemplo de varias condiciones unidas por O: Recuperar el/los alumno/s cuyo apellido sea 'Brown' O cuyo número sea 17.
- Ejemplo de varias condiciones unidas por Y: Recuperar el/los alumno/s de primero Y que estudian Informática.

- e) Ejemplo de consulta a partir de datos que se piden al usuario: Recuperar el/los alumno/s cuya especialidad se indique por teclado.

FORMULARIOS EJEMPLO

Crear un formulario para la tabla SECCIÓN con la siguiente apariencia:

Identificador grupo	Código curso	Semestre
85	MATE2410	Otoño

Año: 1999 Profesor: King

Registro: 1 de 6

- Deberá facilitar la introducción de los datos mediante el uso de cuadros combinados en los campos Código Curso y Semestre.
- En cuanto a sus propiedades generales, el formulario:
 - no deberá dejar ni editar ni eliminar registros, pero sí agregar nuevos.
 - podrá ser minimizable, pero no maximizable.

Crear un formulario para la tabla INFORME_CALIFICACIONES con la siguiente apariencia:

Código alumno: 8

Calificación: A

Registro: 1 de 9

Buscar

Incluir este último formulario como subformulario del formulario anterior, de manera que para cada grupo (sección) se puedan ir viendo las notas de los alumnos de ese grupo.

FORMULARIOS Y SUBFORMULARIOS

Crear formularios que nos permitan:

1. Introducir/ visualizar los datos de un alumno y a continuación sus calificaciones.

The screenshot shows a web application window titled "PRÁCTICA 2 FORMULARIO 1". It contains several input fields for student information: "Nombre" (JUAN), "Código alumno" (22), "Año" (1), and "Especialidad" (CS). Below these fields is a table titled "INFORME_CALIFICACIONES". The table has three columns: "Código alumno", "Identificador sección", and "Calificación". It contains three rows of data: (22, 112, A), (22, 135, A), and a row with a asterisk and 22. At the bottom, there are navigation controls showing "Registro: 1 de 2" and "Sin filtro".

Código alumno	Identificador sección	Calificación
22	112	A
22	135	A
* 22		

Crear cuadros combinados para los campos: Año, especialidad y calificación.

2. Introducir/visualizar los datos de un curso y sus respectivas secciones.

The screenshot shows a web application window titled "PRÁCTICA 2 FORMULARIO 2". It features a header "ASIGNATURAS Y GRUPOS" and a form for course data: "Nombre curso" (Matemáticas discretas), "Código curso" (MATE2410), "Créditos" (3), and "Departamento" (MATE). Below this is a sub-form titled "Grupos de la asignatura" with fields for "Identificador sección" (85), "Semestre" (Otoño), "Año" (1999), and "Profesor" (King). Navigation controls at the bottom show "Registro: 1 de 2" and "Sin filtro".

Identificador sección	Semestre	Año	Profesor
85	Otoño	1999	King

AÑADIR DATOS A LA BD MEDIANTE FORMULARIOS

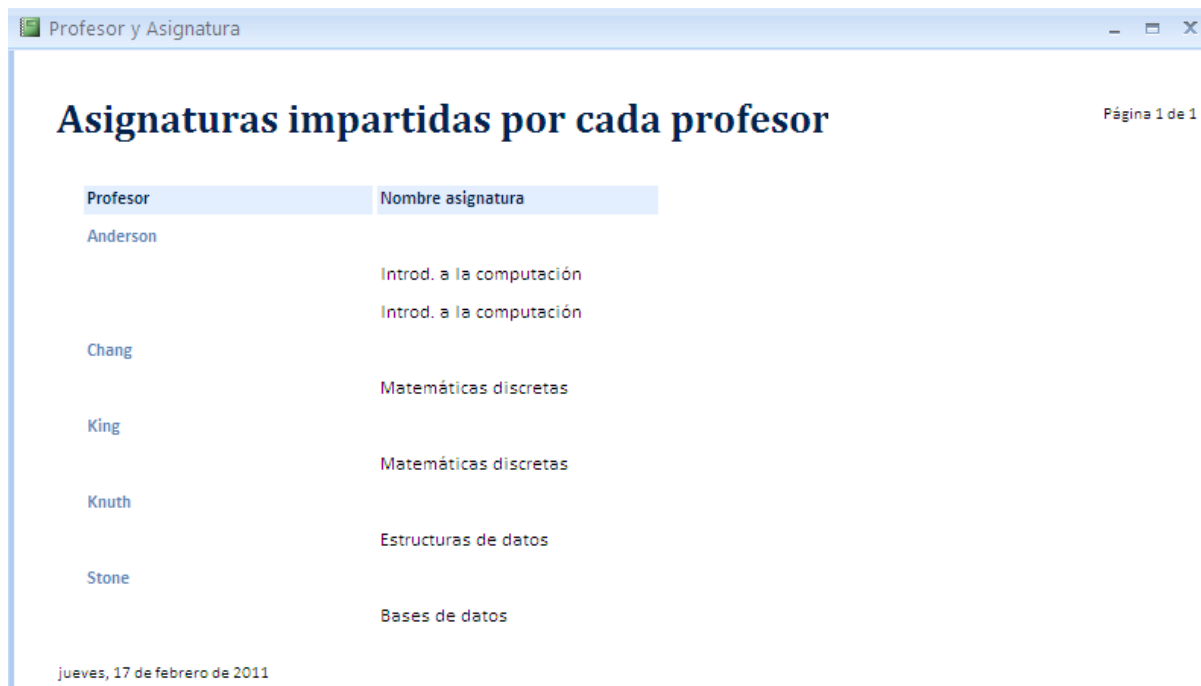
3. **AÑADIR** en las tablas de la BD **LOS DATOS** (filas) **QUE FALTEN** para poder obtener exactamente los informes que aparecen en la siguiente hoja.

ALUMNO					
	Nombre ▾	Código alumno ▾	Año ▾	Especialidad ▾	Agregar nuevo campo
+	Brown	8	3	CS	
+	Juan	12	3	IT	
+	Smith	17	1	CS	
+	Pepe	21	1	CS	
+	Juan	22	1	CS	
+	Pepe	23	3	IT	
+	Ana	99	1	IT	
Registro: 1 de 7 Sin filtro Buscar					
INFORME_CALIFICACIONES					
	Código alumno ▾	Identificador sección ▾	Calificación ▾	Agregar nuevo campo	
	8	85	A		
	8	92	A		
	8	102	B		
	8	135	A		
	17	112	B		
	17	119	C		
	22	112	A		
	22	135	A		
	23	112	D		
Registro: 1 de 9 Sin filtro Buscar					

Ayuda para realizar los informes:

- Crear el diseño por defecto con:
Pestaña *Crear* → *Asistente para informes*
Y luego modificarlo hasta obtener el informe deseado.

1. **EJEMPLO:** Crear un informe que genere las asignaturas impartidas por cada profesor.



Profesor y Asignatura

Asignaturas impartidas por cada profesor

Página 1 de 1

Profesor	Nombre asignatura
Anderson	Introd. a la computación
	Introd. a la computación
Chang	Matemáticas discretas
King	Matemáticas discretas
Knuth	Estructuras de datos
Stone	Bases de datos

jueves, 17 de febrero de 2011

Ayuda para este informe:

- Con Autoformato en Pasos, Vertical y Oficina
- Indicar Agrupamiento y ordenación
- Entender encabezado y pie en cada página frente a encabezado y pie del documento
- Número de página en el encabezado
- Ancho: 19 cm

2. Generar el siguiente informe:

CALIFICACIONES DE LOS ALUMNOS					
CALIFICACIONES DE LOS ALUMNOS				17/02/2011 12:17:50	
Especialidad	Código alumno	Nombre	Curso	Grupo	Nota
CS	8	Brown	3		85 A
					92 A
					102 B
					135 A
	17	Smith	1		112 B
					119 C
IT	22	Juan	1		112 A
					135 A
	23	Pepe	3		112 D

Página 1

3. Crear un informe que genere los grupos (secciones) impartidos por cada profesor.

Grupos por profesor

Profesor	Identificador grupo
Anderson	92
	119
Chang	112
King	85
Knuth	102
Stone	135

4. Generar el siguiente informe pidiendo previamente el nombre del profesor por teclado:

Grupos del profesor Anderson jueves, 17 de febrero de 2011

Profesor
Anderson

Grupos impartidos

92

119

SUBINFORME

Relación de grupos y asignaturas

85	Matemáticas discretas
92	Introd. a la computación
102	Estructuras de datos
112	Matemáticas discretas
119	Introd. a la computación
135	Bases de datos

El subinforme deberá imprimirse solamente una vez y al final del documento.

MACROS (NO HACER)

1. Crear una macro que visualice y permita modificar el primer formulario.
2. Crear una macro que visualice pero no permita modificar el segundo formulario.
3. Abra automáticamente la vista previa de los dos últimos informes del apartado anterior.

PÁGINAS WEB EN ACCESS (NO HACER)

1. Crear una página web para cada una de las tablas de la BD y una página principal que permita el acceso a las mismas.

PRÁCTICA P3

MS ACCESS (3):

Más consultas, importación de datos e integridad referencial

OBJETIVOS

Para realizar esta práctica se necesita:

- No hay actividades previas.

Al terminar esta práctica se debe saber:

- Crear ficheros de Excel y de texto.
- Importar datos desde otros programas, en concreto desde Excel y desde ficheros de texto.
- Crear vínculos con tablas de otras bases de datos.
- Realizar consultas sencillas sobre la base de datos utilizando el lenguaje gráfico QBE (Query By Example). Este lenguaje está implementado en la Vista diseño de la consulta.

NOTA: Existe ayuda sobre las tareas a realizar en esta práctica al final de la práctica P4.
--

IMPORTACIÓN Y VÍNCULOS ENTRE TABLAS DE DISTINTAS BD

1. Crear con los datos correspondientes a las tablas del mismo nombre en la BD UNIVERSIDAD (de la Práctica 1):
un fichero de texto,
➤ ALUMNO.TXT
una hoja en un fichero de Excel:
➤ INFORME_CALIFICACIONES
2. Importar todo a una nueva base de datos Access llamada UNI2_<mi primer apellido>.
3. Crear una nueva BD llamada UNI PROFE_<mi primer apellido> que contenga:
 - Una nueva tabla llamada NOTAS DE PRÁCTICAS con 3 campos: Código alumno, Número de práctica y Nota de la práctica (entre 0 y 10). Establecer su clave principal.
 - Otras dos tablas llamadas ALUMNO e INFORME_CALIFICACIONES que deberán ser las mismas (vinculadas) que las de la BD UNI2_<mi primer apellido> del apartado anterior.
4. Relacionar la nueva tabla NOTAS DE PRÁCTICAS con la vinculada ALUMNO.
¿De qué tipo es la relación? ¿Por qué?
5. Modificar algo en la tabla vinculada y en la original ALUMNO ¿Qué ocurre?

IMPORTACIÓN DE DATOS: BD REPRES_VENTAS

- Importar los campos y datos del libro de Excel **REPRES_VENTAS.XLS** a una nueva BD llamada REPRES_VENTAS_<mi primer apellido>.
- Los datos que deben quedar en las tablas después de la importación aparecen en las siguientes páginas de esta práctica. El tipo de todos los campos en cuyos valores aparece la palabra *euros* es número real (doble), pero visualizados con un formato personalizado (que es el que escribe la palabra euros detrás del número).
- Establecer las claves primarias y relaciones para las tablas de esta nueva BD REPRES_VENTAS_<mi primer apellido>.

ESTADO DE LA BASE DE DATOS REPRES_VENTAS

EMPLEADOS

num-emp	nombre	edad	oficina	titulo	contrato	jefe	cuota	ventas
101	Antonio Viguer	45	12	representante	20-oct-86	104	3.000,00 euros	3.050,00 euros
102	Alvaro Jaumes	48	21	representante	10-dic-86	108	3.500,00 euros	4.740,00 euros
103	Juan Rovira	29	12	representante	1-mar-87	104	2.750,00 euros	2.860,00 euros
104	José González	33	12	dir ventas	19-may-87	106	2.000,00 euros	1.430,00 euros
105	Vicente Pantalla	37	13	representante	12-feb-88	104	3.500,00 euros	3.680,00 euros
106	Luis Antonio	52	11	dir general	14-jun-88		2.750,00 euros	2.990,00 euros
107	Jorge Gutierrez	49	22	representante	14-nov-88	108	3.000,00 euros	1.860,00 euros
108	Ana Bustamante	62	21	dir ventas	12-ene-89	106	3.500,00 euros	3.610,00 euros
109	María Sunta	31	11	representante	12-oct-99	106	3.000,00 euros	3.920,00 euros
110	Juan Víctor	41		representante	13-ene-90	104		760,00 euros

CLIENTES

numclie	nombre	repclie	limitecredito
2101	Luis García Antón	106	650,00 euros
2102	Alvaro Rodriguez	101	650,00 euros
2103	Jaime Llorens	105	500,00 euros
2105	Antonio Canales	101	650,00 euros
2106	Juan Suarez	102	650,00 euros
2107	Julián López	110	350,00 euros
2108	Julia Antequera	109	550,00 euros
2109	Alberto Juanes	103	250,00 euros
2111	Cristóbal García	103	500,00 euros
2112	María Silva	108	500,00 euros
2113	Silvia Marón	104	200,00 euros
2114	Cristina Bulini	102	200,00 euros
2115	Vicente Martínez	101	200,00 euros
2117	Carlos Tena	106	350,00 euros
2118	Junípero Alvarez	108	600,00 euros
2119	Salomón Bueno	109	250,00 euros
2120	Juan Malo	102	500,00 euros
2121	Vicente Ríos	103	450,00 euros
2122	José Marchante	105	300,00 euros
2123	José Libros	102	400,00 euros
2124	Juan Bolto	107	400,00 euros

OFICINAS

oficina	ciudad	region	dir	objetivo	ventas
11	Valencia	este	106	6.760,00 euros	6.990,00 euros
12	Alicante	este	104	8.000,00 euros	7.350,00 euros
13	Castellón	este	105	3.500,00 euros	3.680,00 euros
21	Badajoz	oeste	108	7.250,00 euros	8.360,00 euros
22	A Coruña	oeste	101	1.000,00 euros	1.000,00 euros
23	Madrid	centro	108	2.500,00 euros	1.500,00 euros
26	Pamplona	norte			
28	Valencia	este		9.000,00 euros	

PRODUCTOS	idfab	idproducto	descripcion	precio	existencias
	aci	41001	arandela	0,58 euros	277
	aci	41002	bisagra	0,80 euros	167
	aci	41003	art t3	1,12 euros	207
	aci	41004	art t4	1,23 euros	139
	aci	4100x	junta	0,26 euros	37
	aci	4100y	extractor	28,88 euros	25
	aci	4100z	mont	26,25 euros	28
	bic	41003	manivela	6,52 euros	3
	bic	41089	rodamiento	2,25 euros	78
	bic	41672	plato	1,80 euros	0
	fea	112	cubo	1,48 euros	115
	fea	114	cubo	2,43 euros	15
	imnn	773c	reostato	9,75 euros	28
	imnn	775c	reostato 2	14,25 euros	5
	imnn	779c	reostato 3	18,75 euros	0
	imnn	887h	caja clavos	0,54 euros	223
	imnn	887p	perno	0,25 euros	24
	imnn	887x	manivela	4,75 euros	32
	qsa	xk47	red	3,55 euros	38
	qsa	xk48	red	1,34 euros	203
	qssa	xk48a	red	1,17 euros	37
	rei	2a44g	pas	3,50 euros	14
	rei	2a44l	bomba l	45,00 euros	12
	rei	2a44r	bomba r	45,00 euros	12
	rei	2a45c	junta	0,79 euros	210

PEDIDOS	codigo	numpedido	fechapedido	clie	rep	fab	producto	cant	importe
	1	1100036	2-ene-02	2107	110	aci	4100z	9	225,00
	2	1100036	2-ene-02	2117	105	rei	2a44l	7	315,00
	3	112963	10-may-02	2103	105	aci	41004	28	32,76
	4	112968	11-ene-95	2102	101	aci	41004	34	39,78
	5	112975	11-feb-02	2111	103	rei	2a44g	6	21,00
	6	112979	12-oct-94	2114	108	aci	4100z	6	150,00
	7	112963	10-may-02	2103	105	aci	41004	6	7,02
	8	112987	1-ene-02	2103	105	aci	4100y	11	275,00
	9	112989	10-dic-02	2101	106	fea	114	6	14,58
	10	112992	15-abr-95	2118	108	aci	41002	10	7,60
	11	112993	10-mar-02	2106	102	rei	2a45c	24	18,96
	12	112997	4-abr-02	2124	107	bic	41003	1	6,52
	13	113003	5-feb-02	2108	109	imnn	779c	3	56,25
	14	113007	1-ene-02	2112	108	imnn	773c	3	29,25
	15	113012	5-may-02	2111	105	aci	41003	35	37,45
	16	113013	6-ago-02	2118	108	bic	41003	1	6,52
	17	113024	4-jul-02	2114	108	qsa	xk47	20	71,00
	18	113027	5-feb-02	2103	105	aci	41002	54	41,04
	19	113034	5-nov-02	2107	110	rei	2a45c	8	6,32
	20	113042	1-ene-02	2113	101	rei	2a44r	5	225,00
	21	113045	2-jul-02	2112	108	rei	2a44r	10	450,00
	22	113048	2-feb-02	2120	102	imnn	779c	2	37,50
	23	113049	4-abr-02	2118	108	qsa	xk47	2	7,75
	24	113051	6-jul-02	2118	106	qsa	xk47	4	14,20
	25	113055	1-abr-95	2108	101	aci	4100x	6	1,50
	26	113067	1-nov-02	2111	103	aci	4100x	24	6,00
	27	113058	4-jul-94	2108	109	fea	112	10	14,80
	28	113062	4-jul-02	2124	107	bic	41003	10	24,30
	29	113065	3-jun-02	2106	102	qsa	xk47	6	21,30

Realizar las siguientes consultas sobre la BD REPRES_VENTAS_<mi primer apellido>.accdb

❖ Consultas **PARA PROBAR** (no hace falta guardarlas). Indicar **QUÉ SE OBTENDRÍA** con cada uno de los siguientes criterios de ejemplo: (cada uno es un criterio independiente). Mostrar **TODOS** los campos de la tabla involucrada en cada caso.

1. En el campo **Nombre** de la tabla de **Cientes** tecleemos lo siguiente:
 - Criterio: Ju*
 - Criterio: [INTRODUZCA UN NOMBRE]
 - Criterio: NoEs "JU*"
2. En el campo **Fechapedido** de la tabla **Pedidos**, tecleemos lo siguiente:
 - Criterio: #05/02/02#
 - Criterio: 2/1/2002 o 2/2/2002 o 5/2/2002
 - Criterio: Entre #01/02/02# y #28/02/02#
 - Criterio: >= #01/01/02# y <=#31/12/02#
3. En el campo **Importe** de la tabla **Pedidos** tecleemos lo siguiente:
 - Criterio: >200
 - Criterio: >200 y <=300

❖ Consultas **PARA HACER** y guardar

1. Fabricantes del producto 'red'.
2. Fabricantes de cubos o manivelas.
3. Pedidos del año 2002.
4. Pedidos realizados en la segunda mitad del 2002 (desde junio a diciembre incluidos).
5. Pedidos cuyo importe es superior a 200 euros.
6. Pedidos cuyo importe está comprendido entre 200 y 300 euros.
7. Nombre, apellidos y pedidos del cliente 2103.
8. Fabricantes de manivelas y pernos. (con los datos que hay en la BD, debe dar como solución el fabricante imnn).
9. Pedidos realizados en la fecha que se dé por teclado. Mensaje: INTRODUCIR FECHA.
10. Nombre, apellidos y pedidos del cliente cuyo número se indique por teclado. (probarlo con el cliente 2103 por ejemplo)

Crear una BD vacía llamada BANCO_<primerapellido>.acddb. Importar en ella los datos de la BD BANCO, que se encuentran en el fichero de Excel **BANCO.XLS**.

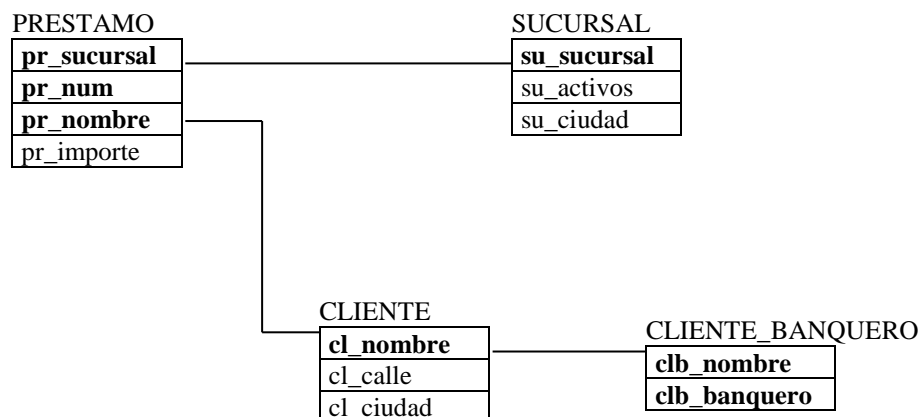
Dado el siguiente esquema, correspondiente a parte de una BD para un Banco:

SUCURSAL (**SU_SUCURSAL**, SU_ACTIVOS, SU_CIUDAD)
 CLIENTE (**CL_NOMBRE**, CL_CALLE, CL_CIUDAD)
 PRESTAMO (**PR_SUCURSAL**, **PR_NUM**, **PR_NOMBRE**, PR_IMPORTE)
 CLIENTE_BANQUERO (**CLB_NOMBRE**, **CLB_BANQUERO**)

Se pide:

1. Seleccionar los préstamos de la sucursal 'PERRYRIDGE'.
2. Seleccionar los préstamos con un importe mayor de 1200 euros.
3. Seleccionar los préstamos de la sucursal 'PERRYRIDGE' con un importe mayor de 1200 euros.
4. Seleccionar los préstamos de la sucursal 'PERRYRIDGE' o con un importe mayor de 1200 euros.
5. Seleccionar las direcciones de los clientes cuyo banquero es 'JOHNSON'.
6. Obtener el nombre de los clientes con préstamos en alguna sucursal de la ciudad de 'HORSENECK'.
7. Obtener el nombre y ciudad de los clientes que tienen al menos un préstamo. Fijarse que existe una persona (Jackson) con un préstamo que no es cliente ¿Cómo podríamos hacer para que salga en la solución con la ciudad en blanco?

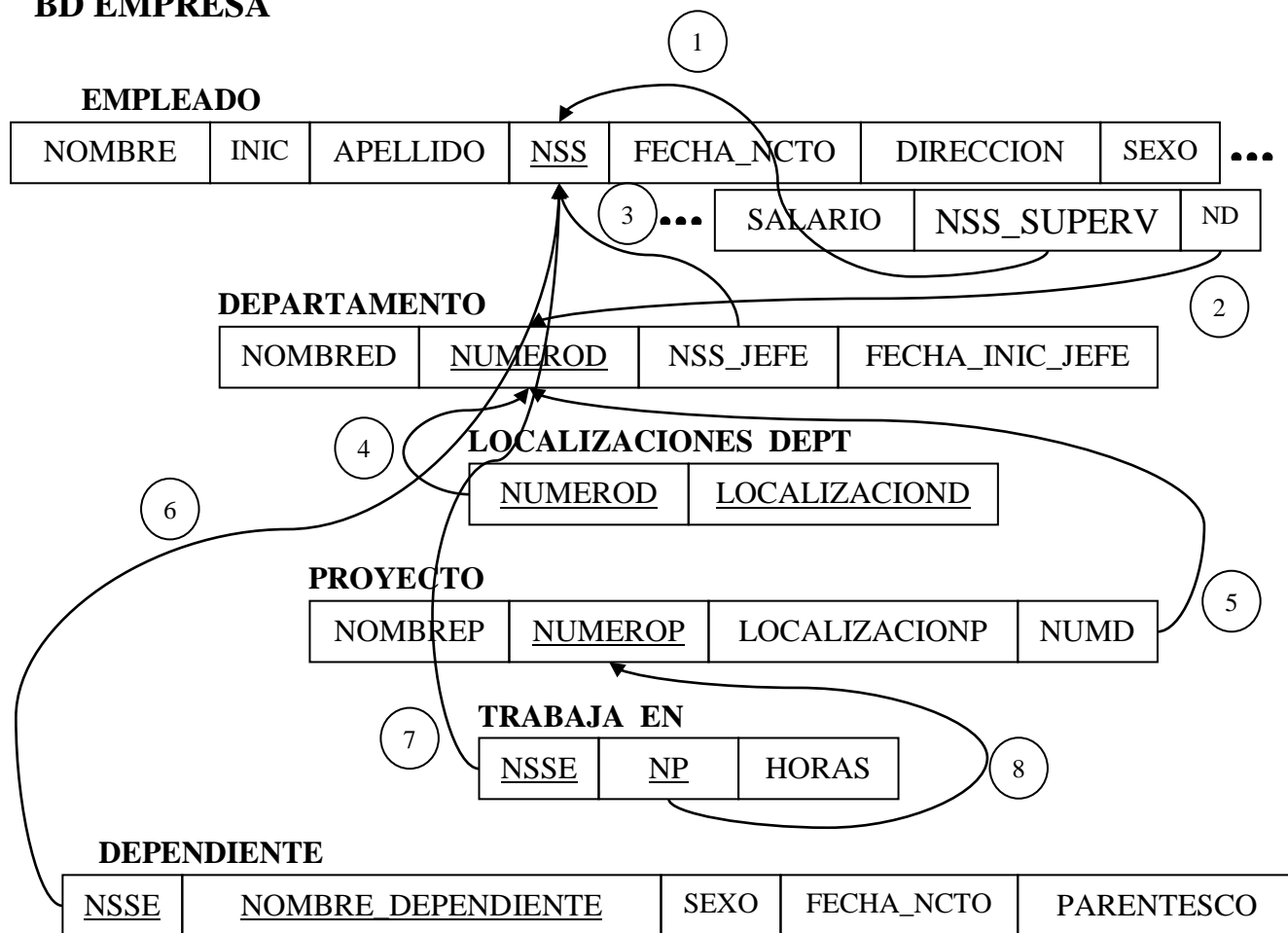
RELACIONES



IMPORTACIÓN, CLAVES PRIMARIAS Y RELACIONES DE LA BD EMPRESA

1. Importar desde el fichero de Excel **EMPRESA.XLS** las tablas para una nueva BD llamada EMPRESA_<mi primer apellido>. Es la misma BD de la Teoría.
2. Adecuar los tipos de datos de los distintos campos de las tablas.
3. Establecer las claves y relaciones de la BD.

BD EMPRESA



ESTABLECIMIENTO DE LAS REGLAS DE INTEGRIDAD REFERENCIAL, ACTUALIZACIONES Y BORRADOS EN CASCADA

- Establecer las reglas de integridad referencial de las BDs :
 - REPRES_VENTAS_<mi primer apellido>
 - EMPRESA_<mi primer apellido>
- Exigir integridad referencial en todos los casos.
- Establecer actualizaciones y borrados en cascada donde consideres oportuno.

Nota: Antes de exigir la integridad referencial conviene verificar que coinciden los tipos en los datos que se quieren relacionar. Si no coinciden habrá que modificar el diseño para que lo hagan.

Indica aquí los borrados en cascada para la BD EMPRESA de cada relación que aparece numerada (dentro del círculo) en la figura de la página anterior:

<u>Relación nº</u>	<u>Borrado en cascada (Sí / No)</u>
--------------------	-------------------------------------

- | | |
|----|--|
| 1. | |
| 2. | |
| 3. | |
| 4. | |
| 5. | |
| 6. | |
| 7. | |
| 8. | |

PRÁCTICA P4

MS ACCESS (4):

Creación y carga de datos en una BD con integridad referencial

OBJETIVOS

Para realizar esta práctica se necesita:

- El esquema E-R de la BD BUQUES y también el esquema relacional correspondiente (creado en las clases de teoría).
- No hay actividades previas.

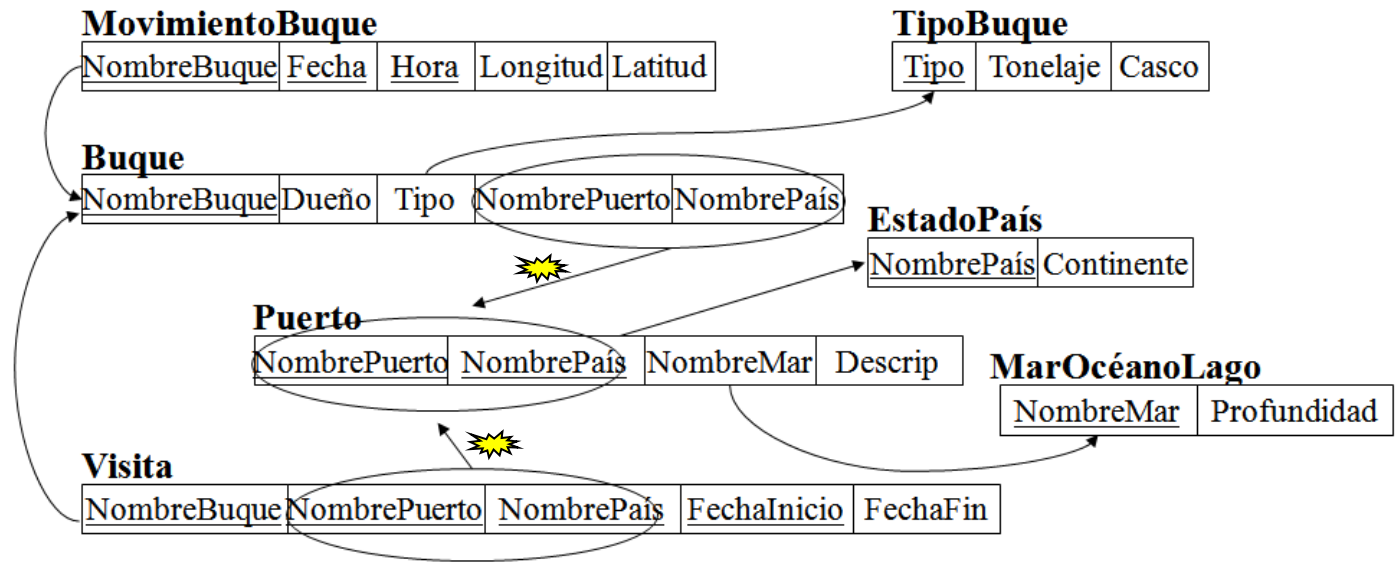
Al terminar esta práctica se debe saber:

- Reconocer claves primarias y claves externas.
- Los distintos tipos de relaciones (vínculos) entre las tablas.
- Cuándo y cómo exigir la integridad referencial.
- Cuándo y cómo establecer actualización y/o borrado en cascada.

NOTA: Existe ayuda sobre las tareas a realizar al final de esta práctica.
--

ESQUEMA RELACIONAL DE LA BD BUQUES

BD BUQUES



TAREAS 1-5 PARA LA BD BUQUES

En esta práctica **SE DEBEN REALIZAR LAS 5 TAREAS EN EL ORDEN QUE SE INDICA**. Para la realización de cada una de estas tareas existe ayuda, detrás de este apartado en “Documentación de apoyo para la práctica 4”.

Tarea 1. Creación de una BD. Apertura de una BD existente

Cambia el nombre de la BD buques00.acddb que se te proporcionará a *BUQUES_<mi primer apellido>*. A partir de este momento se trabajará todo el tiempo con esta BD.

Tarea 2. Creación de tablas


Crea en la BD las tablas del esquema relacional anterior que falten por definir (Buque y Visita). A cada atributo asígnale el tipo de datos más adecuado entre los disponibles. Define las claves primarias que faltan. La **tarea termina** cuando se hayan definido todos los atributos de las tablas del esquema y la clave primaria de cada tabla.

Tarea 3. Establecimiento de reglas de integridad (RI) referencial

La integridad referencial garantiza la ausencia de referencias a tuplas inexistentes. Cada RI referencial se define entre una clave extranjera y la clave primaria a la que hace referencia. Se define una RI referencial por cada clave extranjera. Al definir la RI referencial se determina la reacción del SGBD ante violaciones de la RI producidas por borrados y modificaciones sobre dichas referencias.

Define las RI referenciales que faltan en la BD anterior siguiendo el siguiente criterio:

- Definir borrados y modificaciones **en cascada** para claves extranjeras formadas por un solo atributo.

- No definir borrados ni modificaciones en cascada para claves extranjeras formadas por la combinación de varios atributos (pero sí la restricción de integridad). Las que tienen un  al lado de la flecha.

La **tarea termina** cuando se hayan definido todas las correspondencias entre clave extranjera y la clave primaria a la que hace referencia, que aparecen en el esquema relacional dado.

Tarea 4. Comprobación de las consecuencias de la definición de reglas de integridad.

Para realizar estas comprobaciones se introducen datos de prueba por teclado.

4.1. Comprobación del mantenimiento de la integridad referencial.

Intentar introducir en la tabla Buque el buque llamado **Bilbao**, cuyo dueño sea **Onassis**, de tipo **10** y su puerto original **Hendaya (France)**.

*El sistema tiene que dar un error (podemos salir pulsando **ESC**). ¿Por qué?*

Introducir el Tipo de Buques **10**, que son barcos de **3500Tm** y casco de **FibraVidrio**.

Si intentamos introducir ahora el Buque anterior, ¿funcionará?

Inserta las tuplas necesarias para que se pueda insertar la tupla de la tabla Buque.

Comprobarás que no se pueden insertar los datos en cualquier orden. La tarea 5 trata sobre este tema.

4.2. Comprobación de la modificación en cascada.

Modificar el nombre del mar que se acaba de introducir en la tabla Mares por **Atlántico** (si es el que tienes puesto, cámbialo, por ejemplo, por **Cantábrico**).

Fíjate en la tabla Puerto. ¿Qué ha pasado?

El valor del atributo con el nombre del mar también se ha modificado.

A esto se le llama modificación en cascada. Cuando se modifica el valor de la clave principal, se modifica también el valor de las claves extranjeras que hacen referencia a dicha clave.

*En cambio, no podemos realizar el cambio del nombre del País **France** a **Francia**. ¿Por qué? (tiene que ver con una decisión de la tarea anterior).*

4.3. Comprobación del borrado en cascada.

Borra el registro de la tabla TipoBuque

Además de las modificaciones, también se pueden propagar los borrados. Al borrar el registro de TipoBuque, desaparece el valor 10 de su clave principal, y por tanto se borrarán automáticamente las tuplas que tienen una clave extranjera que referencie a esa clave (en nuestro caso el buque que habíamos insertado) ¿es así? ¿Qué pasará si se borra ahora el país que había almacenado? ¿Queda algún dato más?

Borra ahora todos los datos de la BD, de manera que ésta quede vacía

Tarea 5. Importación de los datos en un orden que no viole ninguna RI referencial.

Esta tarea consiste en importar datos para llenar las tablas de la base de datos. Los datos están almacenados en ficheros de texto.

Pestaña Datos Externos – Area Importar – Icono Archivos de texto

Algún formato de fichero puede ser distinto al definido en las tablas. Además, los datos contenidos en los ficheros de texto pueden estar en distintos idiomas, y las fechas y horas pueden tener formatos personalizados. Si es así soluciona el problema. Cada fichero corresponde a una tabla y tiene el mismo nombre que se les ha dado en el esquema relacional.

Como se ha dicho, no se pueden introducir los datos en cualquier orden. Queremos obtener un orden que permita importar todos los ficheros de modo que se cumplan las reglas de integridad referencial.

La tarea 5 finaliza cuando:

- Se conozca un orden de importación de los datos de las diferentes tablas que no viole ninguna RI.
- Se hayan importado los datos de todas las tablas.


➤ Indica el orden en el que has importado las tablas.

Tarea 1. Creación de la Base de Datos (BD)


Para crear una BD nueva basta con ejecutar Microsoft Access. El menú inicial consta de las siguientes opciones:

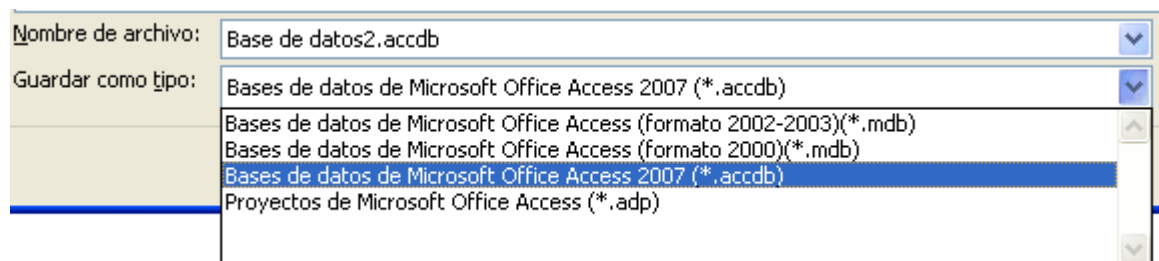
- **Base de Datos en blanco:** opción a elegir para esta tarea.
- **Abrir una BD existente:** opción a utilizar en sucesivas sesiones de trabajo con una BD ya creada.


Creación de una base de Datos nueva en ACCESS

Si no sale la parte derecha, darle al icono Restaurar  de la ventana.



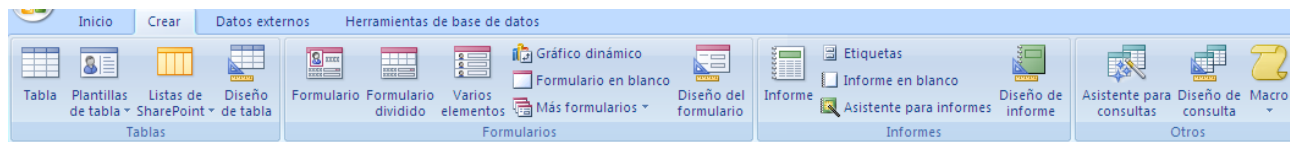
Pinchamos en el icono de la carpeta  al lado del “nombre de archivo” para que sea compatible con otras versiones anteriores o para guardarlo en una carpeta diferente:



También podemos pasar una BD de una versión a otra anterior pinchando en el icono de Office  (esquina superior izquierda) y seleccionando Guardar Como... (La opción Convertir... solamente convierte una base de datos anterior a otra en el formato de la versión actual).

Tarea 2. Creación de las tablas

Tras abrir la BD, se pueden crear las tablas.



Para crear una nueva tabla elegir la pestaña **Crear** icono **Diseño de tabla**. Aparece una tabla de tres columnas. La primera contiene los nombres de los atributos de la tabla. La segunda su tipo de datos y la tercera una breve descripción de cada atributo.

Introducir los datos de los atributos de una tabla. Al terminar con la tabla, cerrar la ventana. El sistema nos preguntará cuál el nombre de la tabla. Introducir del mismo modo las demás tablas de la BD.

Los tipos disponibles en cada SGBD's tienen diferentes denominaciones. Suelen cubrir un espectro de dominios parecido: *strings*, *fechas*, *números*, etc. A continuación se incluye un resumen de los tipos disponibles en Access. La información sobre el tamaño que ocupa cada una de las opciones se puede consultar en la ayuda del sistema.

Tabla 1. Tipos de datos en Access 2016

Tipo	Valores admitidos
Texto corto (Por defecto)	Texto o combinación de texto y números. No se podrán realizar cálculos con los números (ej. números de teléfono)
Texto largo (antes Memo)	Texto extenso, o combinación extensa de texto y números.
Número	Datos numéricos utilizados en cálculos matemáticos. Para obtener más información sobre la forma de establecer un tipo numérico específico, ver la ayuda sobre <i>propiedad Tamaño del campo</i> .
Fecha/Hora	Valores de fecha y hora para los años del 100 al 9999.
Moneda	Valores de moneda y datos numéricos utilizados en cálculos matemáticos en los que estén implicados datos que contengan entre uno y cuatro decimales. La precisión es de hasta 15 dígitos a la izquierda del separador decimal y hasta 4 dígitos a la derecha del mismo.
Autonumeración	Número secuencial (por defecto incrementado de uno en uno) único, o número aleatorio que Access asigna cada vez que se agrega una nueva tupla a una tabla. Los atributos Autonuméricos no se pueden actualizar. Más información en la ayuda sobre la propiedad <i>Nuevos valores</i> .
Sí/No	Valores Sí ; y No , y atributos que contengan uno de entre un par valores (Sí/No, Verdadero/Falso o Activado/desactivado).
Objeto OLE	Objeto (como por ejemplo una hoja de cálculo de Microsoft Excel, un documento de Microsoft Word, gráficos, sonidos u otros datos binarios) vinculado o incrustado en una tabla de Microsoft Access.
Hipervínculo	Texto o combinación de texto y números almacenada como texto y utilizada como dirección de hipervínculo (dirección URL).
Asistente para búsquedas	Crea un campo que permite elegir un valor de otra tabla o de una lista de valores mediante un cuadro de lista o un cuadro combinado. Al hacer clic en esta opción se inicia el Asistente para búsquedas, que crea un campo de búsqueda. Al salir del Asistente, Microsoft Access establece el tipo de datos basándose en los valores seleccionados en él.

Los tipos disponen de una serie de propiedades que permiten adecuarlos a nuestras necesidades. A continuación se comenta la mayor parte de ellas:

- **Valor Predeterminado.** Valor introducido automáticamente en el atributo cuando se agrega una tupla. Por ejemplo, en la tabla Direcciones se puede establecer *San Sebastián* como valor predeterminado del atributo Ciudad. Cuando se agregue una tupla a la tabla, se puede aceptar este valor o introducir otra ciudad.
- **Regla de validación y texto de validación.** Para especificar requisitos de los datos introducidos en una tupla. El *texto de validación* contiene un mensaje que se mostrará al usuario si se introducen datos que infringen la *regla de validación*. Esta propiedad permite introducir reglas de integridad de dominio.
- **Requerido.** Para especificar si es o no obligatorio un valor distinto de nulo para el atributo en cada tupla (cuando el valor de esta propiedad es **sí**).

Por ejemplo, si queremos que el atributo Apellidos tenga un valor en cada tupla. Si deseamos permitir el valor Nulo en un campo, además de establecer **no** para esta propiedad, si hemos puesto algún valor en la propiedad *regla de validación*, deberemos incluir "*regla_de_validación O Es Nulo*". Esta propiedad no se aplica a atributos de tipo Autonumérico.

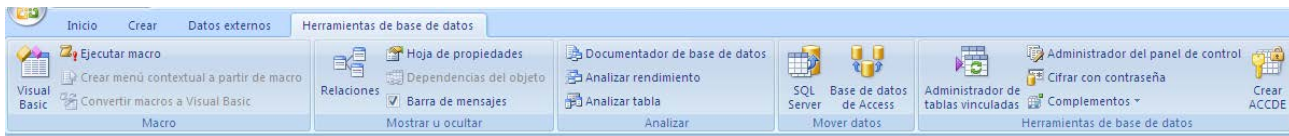
- **Indexado.** Se puede usar para establecer un índice de un solo atributo. Los índices hacen que las consultas basadas en los atributos indexados sean más rápidas. También aceleran las operaciones de ordenación y agrupación. Por ejemplo, si se buscan empleados basándose en un atributo llamado *Apellidos*, se puede crear un índice sobre dicho atributo para hacer más rápida la búsqueda.
- **Formato.** Afecta sólo a cómo se muestran los datos, y no a cómo se almacenan. Sirve para personalizar la forma en la que se muestran e imprimen números, fechas, horas y textos.

Por ejemplo, para un campo *Precio*, se puede establecer en que es de tipo *Moneda* y en la propiedad *lugares decimales* a 2 o a Auto. Si alguien introduce en una tupla el valor 4321,678, el número se mostrará como 4.321,68 €. Access dispone de formatos predefinidos para los tipos Fecha/Hora, Numérico, Moneda, Texto, y Sí/No.

- **Lugares decimales.** Para indicar el número de cifras decimales a usar para presentar números.
- **Tamaño.** Para establecer el máximo de caracteres almacenados en un campo de tipo Texto, Numérico o Autonumérico. Para el tipo *Texto corto*, es un número entre 0 y 255. El valor por defecto es 255.
- **Máscara de entrada.** Para facilitar la introducción de datos y controlar los valores que se pueden introducir. Por ejemplo, se puede crear una máscara de entrada para un campo *Número de teléfono* que muestre exactamente cómo debe introducirse un nuevo número: (____) ____-____.
- **Permitir Longitud 0.** Para especificar si una cadena de longitud cero ("") es una entrada válida para un atributo de una tupla. Sólo se aplica a atributos de tipo texto o hipervínculo.
- **Nuevos valores.** Para especificar la forma en que se incrementan los atributos de tipo Autonumérico al agregar nuevas tuplas a una tabla. Se aplica únicamente a atributos de tipo Autonumérico.

Tarea 3. Establecimiento de reglas de integridad (RI) referencial

Access permite definir la relación entre clave extranjera y clave primaria. Además se puede especificar el modo en que deberá gestionar la integridad referencial.



Pinchar en la pestaña **Herramientas de base de datos** y seleccionar el icono **Relaciones**. Nos preguntará por las tablas a mostrar. Seleccionar las tablas de interés (en nuestro caso todas). Aparecerá un conjunto de recuadros que representan las tablas seleccionadas. Cada recuadro incluye los nombres de los atributos de la tabla. Si deseamos relacionar atributos de una misma tabla, debemos seleccionar dos copias de la misma tabla y operar como si fueran tablas diferentes.

Para establecer una relación se selecciona uno o varios de los atributos de una tabla y se arrastra el ratón hasta llegar al/los atributo/s de la otra tabla. Es recomendable arrastrar desde la clave principal hacia la clave extranjera. Si se hace al revés, Access puede determinar mal la cardinalidad. Tras esta operación se muestra una tabla donde figuran los datos de la relación que se está intentando establecer. Aquí se puede seleccionar la opción: *Exigir integridad referencial*.

La integridad referencial garantiza que no existirán referencias a tuplas inexistentes. Se puede indicar que se conserve automáticamente la integridad referencial transmitiendo modificaciones y/o borrados en cascada a todas las tuplas afectadas. Esta posibilidad está presente en la opción *Exigir integridad referencial*.

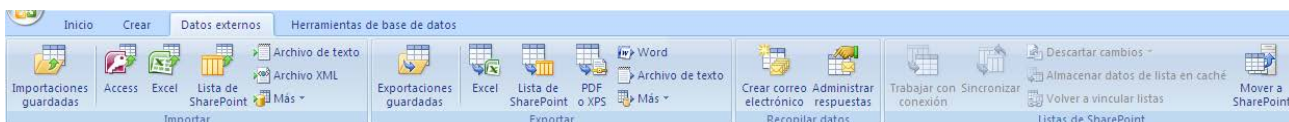
Por último, conviene comprobar que la cardinalidad de cada relación es la adecuada. En cada relación 1:N deberá aparecer un 1 en la parte "1" y un "infinito" en la parte "N". Las relaciones M:N se representan como *dos* relaciones 1:N (si es n-aria, en *n*). La parte "N" de todas ellas corresponde a la tabla auxiliar que hemos creado. Estas operaciones se deben realizar para cada clave extranjera definida.

Tarea 4. Comprobación de las consecuencias de la definición de las RI referencial.

No se incluye ayuda adicional.

Tarea 5. Definición de un orden para importar datos que forma que no se violen las RI referencial. Importación de los datos.

Hay que conocer dónde están los ficheros de datos y su formato. Access 2007 cuenta con un asistente para importar datos. Para invocarlo seleccionar la pestaña **Datos externos** e ir al área **Importar**.



El asistente nos pide que localicemos el fichero desde donde importar los datos. Habrá que pinchar en el icono correspondiente al tipo de fichero del que vamos a traernos los datos (Access, Excel, texto, más...).

El asistente pide algunos datos para realizar la importación con éxito. Por ejemplo, si son ficheros de texto, preguntará si los datos tienen un ancho fijo o si están delimitados por algún carácter (como tabuladores). Tras proporcionar estos datos se importan los datos a una tabla. Tras la importación, nos muestra si se han detectado errores y cuántos registros se han visto afectados.

PRÁCTICA P5

El SGBD MySQL (1): Inicio con XAMPP y MySQL

OBJETIVOS

Para realizar esta práctica se necesita:

- No hay actividades previas.

Al terminar esta práctica se debe saber:

- El funcionamiento básico de MySQL, así como las instrucciones SQL de definición de datos (parte LDD del lenguaje SQL) y de actualización. Este programa nos servirá para practicar el lenguaje SQL (Tema 4 de Teoría).
 - Familiarizarse con el funcionamiento básico de XAMPP y de MySQL.
 - Controlar la creación de bases de datos, de tablas y de relaciones en SQL.
 - Aprender a modificar el diseño de las tablas con SQL.
 - Aprender a insertar, modificar y borrar datos en SQL.
 - Saber guardar las instrucciones hechas y la propia base de datos en un script.

<p>NOTA: TODO lo referente a la descarga y manuales de XAMPP y MySQL está en el egela de BD. Además tenemos comandos y ejemplos en el tema 4 de teoría.</p>
--

PASOS PARA TRABAJAR CON MYSQL Y XAMPP

1. Abrir el panel de control de XAMPP

Inicio → equipo → C → XAMPP → xampp-control.exe

2. Arrancar el Servidor de MySQL desde este panel de XAMPP

MySQL Start

3. Entrar en MySQL desde la línea de comandos

En general, una vez arrancado el servidor de MySQL se entra desde la línea de comandos c:\dir_mysql\bin con:

```
c:\dir_mysql\bin > mysql -h<host> -u<nombre de usuario> -p<password>
```

donde dir_mysql es el directorio en el que está instalado MySQL. La opción -h corresponde al host y no hace falta si estamos en la misma máquina que el servidor. Con la opción -u se indica el nombre del usuario que va a entrar y con -p su password. Si no se indica la password la pedirá con "Enter password: "

Concretamente, para entrar en el MySQL de XAMPP como usuario root (creado por defecto sin contraseña), nos situamos en el directorio BIN de MySQL y entramos:

- CD \XAMPP\MYSQL\BIN
- mysql -uroot

Cuando se entra en MySQL, sale un mensaje de ayuda y el prompt de la línea de comandos cambia a *mysql* >

4. Trabajar con MySQL

Existen dos tipos de **ayuda local** en MySQL:

- Para algunos comandos propios de MySQL y los sitios, tanto locales como de la web, en los que encontrar más ayuda:
mysql> help
- Para acceder al manual de MySQL que incluye la sintaxis y funcionamiento de las instrucciones SQL implementadas:
mysql> help contents

5. Salir de MySQL

mysql > QUIT

6. Parar el servidor de MySQL desde el panel de XAMPP

MySQL Stop

COMANDOS ESPECÍFICOS DE MYSQL

Ver el estado de MySQL (qué usuario somos, qué BD estamos usando, etc...)

mysql > STATUS

Mostrar las BDs existentes

mysql> SHOW DATABASES;

Usar una BD existente (BD actual)

mysql > USE nombre_de_la_BD

Mostrar las tablas de la BD actual

`mysql > SHOW TABLES;`

Mostrar los campos de una tabla

`mysql > DESCRIBE nombre_tabla;`

Mostrar la instrucción de creación de una tabla

`mysql > SHOW CREATE TABLE nombre_tabla;`

Aquí se puede saber el tipo de tabla (por defecto InnoDB) y el charset (es el conjunto de caracteres utilizado por el SGBD, por defecto latin1 – cp1252 de windows ó ISO 8859-1 – para el servidor y la BD, y cp850 – latino específico del DOS – para el cliente y la conexión)

Importar datos desde un fichero de texto y cargarlos en una tabla

`mysql > LOAD DATA INFILE 'nombre_fichero.txt'`
`INTO TABLE nombre_tabla (campo1, campo2,...,campon);`

El fichero de texto a importar ha de estar en el directorio de esa base de datos que es

`C:\dir_mysql\DATA\carpeta_con_el_nombre_de_la_BD\`

Si el fichero de texto a importar está en otro directorio, hemos de poner su ruta completa al estilo UNIX (es decir, separando las distintas carpetas con /). Ejemplo: ‘C:/fichero.txt’

Ejecutar scripts sobre la BD

Los ficheros de script son ficheros de texto que contienen instrucciones en SQL.

Los ficheros de script con instrucciones SQL tienen habitualmente la extensión .sql .

`mysql > SOURCE ruta_completa_del_fichero_de_script`

Ejemplo: `source F:\practicassBD\consult1.sql`

Grabar la sesión en un fichero

`mysql> TEE nombre_fichero.TXT`

Ejemplo: `TEE C:\TEMP\FIC1.TXT`

A partir del momento en que se ejecute este comando, todo lo que aparezca en la pantalla se guarda en el fichero de texto que se ha indicado.

Para **dejar de grabar**: QUIT o NOTEE (Si después ponemos simplemente TEE sigue añadiendo al fichero anterior)

Comentarios en los comandos

Existen dos alternativas para poner comentarios en los comandos de MySQL:

- Dos guiones al principio de la línea:
`mysql> -- Consulta 1`
- También se pueden usar los comentarios al estilo de C y Java en cualquier parte de las instrucciones:
`/* */`

COPIA DE SEGURIDAD DE UNA BD

Para crear una **copia de seguridad** de una BD (se crea un script con todo lo referente a la BD – tablas y datos):

- Opción 1: Desde XAMPP → Arrancar Apache desde XAMPP → Poner en el navegador la URL: **localhost/xampp** → phpMyAdmin → seleccionar la BD → Exportar (esto descarga un fichero con la copia llamado `nombre_de_la_bd.sql` en la carpeta Descargas)
- Opción 2: Desde el directorio de instalación de MySQL → Desde la línea de comandos (fuera de mysql pero en el mismo directorio `c:\xampp\mysql\bin`):
mysqldump –uroot nombre_de_la_BD > nombre_fichero.sql

COMANDOS DEL ESTÁNDAR SQL Y CONVENCIONES UTILIZADAS EN LA SINTAXIS

Los comandos del estándar SQL siguen la sintaxis del estándar SQL92. Por ejemplo, para crear una tabla nueva dentro de la BD actual se sigue la sintaxis de la instrucción CREATE TABLE de SQL92.

Todas las instrucciones acaban con punto y coma (;)

A la hora de indicar la sintaxis de los distintos comandos de SQL se sigue la notación conocida como BNF (Backus Normal Form), que utiliza las convenciones descritas en la siguiente tabla:

SIMBOLO	SIGNIFICADO
Palabra en MAYÚSCULAS	Palabra reservada de SQL
Palabra en minúsculas	A sustituir por lo que sea pertinente
Corchetes []	Opcional de 0 a 1 veces
Llaves { }	Elegir una de las opciones que aparezcan dentro
Barra vertical	Separa las distintas opciones (como un O lógico)
Puntos suspensivos ...	Repetición de lo que aparezca justo antes de los puntos suspensivos

SINTAXIS DE SQL - LDD (Lenguaje de Definición de Datos)

Crear una nueva BD (CREATE DATABASE)

```
CREATE DATABASE [IF NOT EXISTS] db_name
```

db_name es el nombre de la base de datos.

Borrar una BD existente (DROP DATABASE)

```
DROP DATABASE [IF EXISTS] db_name
```

Crear una nueva tabla (CREATE TABLE)

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name  
[(create_definition,...)] [table_options] ;
```

tbl_name es el nombre de la tabla.

create_definition es una de las siguientes opciones:

- a) col_name data_type [NOT NULL | NULL] [DEFAULT default_value] [AUTO_INCREMENT]
- b) [CONSTRAINT symbol] PRIMARY KEY (col_name,...)
- c) [CONSTRAINT symbol] UNIQUE [INDEX] (col_name,...)
- d) [CONSTRAINT symbol] FOREIGN KEY (col_name,...) [reference_definition]
- e) CHECK (expr)

col_name es el nombre de la columna (campo de la tabla).

default_value es el valor a poner por defecto en el campo.

data_type:

```
INT[(length)] [UNSIGNED] [ZEROFILL]
REAL[(length,decimals)] [UNSIGNED] [ZEROFILL]
DOUBLE[(length,decimals)] [UNSIGNED] [ZEROFILL]
FLOAT[(length,decimals)] [UNSIGNED] [ZEROFILL]
DECIMAL(length,decimals) [UNSIGNED] [ZEROFILL]
CHAR(length) [BINARY]
VARCHAR(length) [BINARY]
DATE → Fecha con formato 'YYYY-MM-DD'
TIME → Hora con formato 'HH:MM:DD'
DATETIME → Fecha y hora con formato 'YYYY-MM-DD HH:MM:SS'
TIMESTAMP → Marca de tiempo 'YYYYMMDDHHMMSS'
```

symbol es el nombre para la restricción.

reference_definition:

```
REFERENCES tbl_name [(index_col_name,...)]
    [ON DELETE reference_option]
    [ON UPDATE reference_option]
```

reference_option:

```
RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT
```

expr es la expresión a validar.

table_options:

```
TYPE = { InnoDB | MYISAM }
[[DEFAULT] CHARACTER SET charset_name] [COLLATE collation_name]
```

Cambiar la estructura de una tabla (ALTER TABLE)

```
ALTER TABLE tbl_name alter_spec [, alter_spec ...]
```

alter_spec es una de las siguientes opciones:

- a) ADD [COLUMN] create_definition [FIRST | AFTER col_name]
- b) ADD [CONSTRAINT symbol] PRIMARY KEY (col_name,...)
- c) ADD [CONSTRAINT symbol] UNIQUE (col_name,...)
- d) ADD [CONSTRAINT symbol] FOREIGN KEY (col_name,...)
[reference_definition]
- e) ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}
- f) DROP [COLUMN] col_name
- g) DROP PRIMARY KEY
- h) MODIFY [COLUMN] create_definition [FIRST | AFTER col_name]
- i) table_options

Borrar datos y definición de una tabla (DROP TABLE)

```
DROP [TEMPORARY] TABLE [IF EXISTS] tbl_name [, tbl_name,...] [RESTRICT | CASCADE]
```

SINTAXIS DE SQL – LMD (Lenguaje de Manipulación de Datos): INSTRUCCIONES DE ACTUALIZACIÓN DE DATOS

Insertar nuevos valores en una tabla (INSERT INTO)

Lo típico es utilizar INSERT para introducir una fila de datos nueva con la sintaxis:

```
INSERT [INTO] tbl_name [(col_name,...)]  
VALUES ((expression | DEFAULT),...),(...),...  
[ ON DUPLICATE KEY UPDATE col_name=expression, ... ]
```

expression es la expresión cuyo valor se asigna a la columna correspondiente de la tabla.

INSERT también permite llenar una tabla con el resultado de una SELECT, utilizando esta sintaxis:

```
INSERT [INTO] tbl_name [(col_name,...)]  
SELECT ...
```

Modificar datos de una tabla (UPDATE)

```
UPDATE tbl_name [, tbl_name ...]  
SET col_name1=expr1 [, col_name2=expr2 ...]  
[WHERE where_condition]
```

expr es la expresión cuyo valor se va a asignar a la columna que aparece a la izquierda del igual.

where_condition es la expresión de condición que debe ser cierta (cumplirse) para cada fila a tratar.

Borrar datos de una tabla (DELETE)

```
DELETE FROM tbl_name  
[WHERE where_condition]
```

PROBAR LOS SIGUIENTES COMANDOS EN ORDEN

Antes de empezar, si no existe, se debe **crear una nueva carpeta llamada TEMP** en el directorio raíz de la unidad C:\.

1. Entrar en MySQL desde XAMPP
2. Indicar el comando para que la sesión se guarde en el fichero **C:\TEMP\practica5.txt**
3. Ver las BDs que hay en MySQL
4. Crear una BD llamada EMPRESA_EJEMPLO
CREATE DATABASE EMPRESA_EJEMPLO;
5. Ponerla como BD actual con → **USE EMPRESA_EJEMPLO**
6. Ver el status (configuración de MySQL) con el comando → **STATUS**
7. Ver las tablas que tiene la BD actual con → **SHOW TABLES;**
8. Crear la tabla DEPTO parecida a la de la Tª 4 del tema 4 de teoría. Esta tabla está relacionada con la tabla EMPLEADO a través de su campo NSS_JEFE.

EMPLEADO(NSS, NOMBRE, SUELDO, ND)

DEPTO(NOMBRED, NUMEROD, NSS_JEFE, FECHA_INIC_JEFE)

Primero debemos crear la tabla empleado para poder poner luego la integridad referencial.

```
CREATE TABLE EMPLEADO (
  NSS CHAR(9) NOT NULL, /* cadena longitud fija 9 caracteres, obligatorio */
  NOMBRE VARCHAR(50), /* cadena longitud maxima 50 caracteres */
  SUELDO DECIMAL(10,2), /* numero real con total 10 digitos, 2 de ellos decimales */
  ND INT, /* numero entero */
```

```
PRIMARY KEY(NSS)); /* Clave Primaria de la tabla, campo NSS */
```

```
CREATE TABLE DEPTO (
  NOMBRED VARCHAR(15) NOT NULL,
  NUMEROD INT NOT NULL,
  NSS_JEFE CHAR(9),
  FECHA_INIC_JEFE TIMESTAMP, /* marca temporal que contiene fecha y hora */
```

```
PRIMARY KEY(NUMEROD), /* Clave Primaria, campo numerod */
UNIQUE(NOMBRED), /* Clave Alternativa, campo nombred */
FOREIGN KEY(NSS_JEFE) /* Clave Externa, campo nss_jefe */
REFERENCES EMPLEADO(NSS) /* hace referencia al campo nss de empleado */
ON UPDATE CASCADE /* cambiar el nss_jefe cuando cambie el nss del empleado */
ON DELETE SET NULL); /* dejar el campo nss_jefe vacío, al borrar el empleado de referencia */
```

9. Ver la instrucción SQL de creación de la tabla DEPTO
SHOW CREATE TABLE DEPTO;
10. Ver los campos de la tabla. Fijarse que el tipo timestamp tiene por defecto la fecha y hora actuales.
DESCRIBE DEPTO;

11. Añadir una nueva fila a la tabla depto (fijarse que da un error)
`INSERT INTO DEPTO VALUES ('CALIDAD',25,'123456789','2000-1-1');`
12. Porque antes hemos de añadir al empleado correspondiente con:
`INSERT INTO EMPLEADO(NSS) VALUES('123456789');`
13. Repetir 11 para insertar en depto y ver resultados en las tablas. Ver el contenido (datos) de las tablas con
`SELECT * FROM EMPLEADO;`
`SELECT * FROM DEPTO;`
14. Exigir la integridad referencial entre EMPLEADO.ND y DEPT.NUMEROD con actualización en cascada y cuando se borre un departamento el ND del empleado quede en blanco (null).
`ALTER TABLE EMPLEADO ADD FOREIGN KEY(ND) REFERENCES DEPTO(NUMEROD) ON UPDATE CASCADE ON DELETE SET NULL;`
15. Ver la instrucción de creación de la tabla empleado.
16. Quitar el atributo fecha_inic_jefe
`ALTER TABLE DEPTO DROP FECHA_INIC_JEFE;`
17. Ver los campos de la tabla depto con describe.
18. Volver a ponerlo, pero ahora con tipo que solamente tiene la fecha (date).
`ALTER TABLE DEPTO ADD FECHA_INIC_JEFE DATE;`
19. Ver los campos de la tabla depto con describe.
20. Ponerle como valor por defecto al nuevo campo '2015-1-1'
`ALTER TABLE DEPTO ALTER FECHA_INIC_JEFE SET DEFAULT '2015-01-01';`
21. Ver los campos de la tabla depto con describe.
22. Insertar los valores ('VENTAS',35) con la otra sintaxis
`INSERT INTO DEPTO (NOMBRED, NUMEROD) VALUES ('VENTAS',35);`
23. Ver los datos de la tabla depto con select *.
24. Cambiar la fecha de 'CALIDAD' a la de hoy con current_date.
`UPDATE DEPTO SET FECHA_INIC_JEFE=CURRENT_DATE WHERE NOMBRED='CALIDAD';`
25. Ver los datos de la tabla depto con select *.
26. Crear una copia de seguridad de la BD (usando la opción 1, que se indica en la página 35)
27. Volver a entrar en MySQL
28. Borrar el departamento de Ventas de la BD
`DELETE FROM DEPTO WHERE NOMBRED='VENTAS';`
29. Borrar todos los datos de la tabla
`DELETE FROM DEPTO;`
30. Borrar las tablas. Si no nos deja, primero quitar las claves externas (foreign keys) con ALTER TABLE nombre-tabla DROP FOREIGN KEY nombre-restriccion →
`ALTER TABLE EMPLEADO DROP FOREIGN KEY empleado_ibfk_1;`
`DROP TABLE DEPTO;`
31. y borrar la BD.
`DROP DATABASE EMPRESA_EJEMPLO;`
32. Salir de MySQL → EXIT ó QUIT
33. Salir de XAMPP → MySQL Stop y luego Exit

Al finalizar, **GUARDAR PARA ENTREGAR más adelante**, el fichero de la sesión **practica5.txt** (que debería estar en C:\TEMP) y el fichero de la copia de seguridad **empresa_ejemplo.sql** (que estará en Descargas si usamos la opción 1).

PRÁCTICA P6

El SGBD MySQL (2):

LDD y actualización de datos en SQL

OBJETIVOS

Para realizar esta práctica se necesita:

- No hay actividades previas.

Al terminar esta práctica se debe saber:

- El lenguaje LDD de SQL
 - Crear/borrar bases de datos (CREATE/DROP DATABASE)
 - Crear/borrar tablas que soporten la integridad referencial (CREATE/DROP TABLE)
 - Modificar la estructura de una tabla (ALTER TABLE).
- El lenguaje LMD de SQL: Instrucciones de actualización de datos.
 - Insertar registros (INSERT INTO)
 - Modificar registros (UPDATE)
 - Borrar registros (DELETE)

<p>NOTA: Toda la ayuda sobre el manejo de MySQL necesaria para la realización de esta práctica está en la práctica anterior y en el tema 4 de la teoría.</p>

CREAR LA BD BUQUES_<apellido>

1. Crea en SQL una base de datos llamada BUQUES_<apellido> idéntica a la BD BUQUES de la práctica 4 y siguiendo los criterios descritos a continuación:

Nombres de tablas, atributos y valores de datos:

- Como utilizamos el charset `latin1`, NO pondremos caracteres especiales del castellano como acentos, eñes,...(por ejemplo, en vez de ñ poner ni)

Tipos de datos de las tablas:

- Todos los que sean texto los definiremos como cadenas de caracteres de cómo máximo 20 caracteres.
- Todos los que sean números los definiremos como enteros excepto tonelaje, que lo pondremos de tipo real con 7 dígitos de los cuales 2 serán decimales.
- Tipo de dato para las fechas: `TIMESTAMP`
- En los campos que no formen parte de la clave primaria, siempre permitiremos nulos.

Establecimiento de reglas de integridad (RI) referencial de las tablas:

Define las RI referenciales en las tablas de la BD siguiendo el siguiente criterio:

- Definir borrados y modificaciones **en cascada** para claves extranjeras formadas por un solo atributo.
- No definir borrados ni modificaciones en cascada para claves extranjeras formadas por la combinación de varios atributos (pero sí la restricción de integridad)

2. Añade el registro ('AndraMari', 'Santander','España','22/11/2020 0:00:00', '23/11/2020 10:00:00') en la tabla Visita respetando la integridad referencial.
3. Realiza una copia de seguridad de la BD BUQUES_<apellido>.

CREAR LA BD EMPRESA_<apellido> (partiendo de otra ya dada)

1. Descarga de eGela el fichero **scriptsCREACIONbds2020.zip** y descomprímelo.
2. Ejecuta el script **scripts_BDs_2020.sql** desde MySQL, con el comando `SOURCE`.
3. Verifica, con `SHOW DATABASES`, que se haya creado en MySQL una BD llamada EMPRESA (también se han debido crear otras 3: UNIVERSIDAD, REPRES_VENTAS y BUQUES).
4. Crea en SQL una base de datos nueva llamada EMPRESA_<apellido> idéntica a la BD EMPRESA de la práctica 3, partiendo de la BD EMPRESA que se acaba de crear. Pasos a dar:
 - a. Realiza una copia de seguridad de la BD EMPRESA que está en MySQL. A continuación, abre el fichero generado y reemplaza en esa copia de seguridad el motor (ENGINE) `MyISAM` por `InnoDB` para poder poner las foreign keys.
 - b. Crea en MySQL una nueva BD vacía llamada EMPRESA_<apellido> y ponla como BD actual, con el comando `USE`.
 - c. Ejecuta la copia de seguridad (la generada al terminar el apdo. a) en MySQL con el comando `SOURCE`. Al ejecutar este comando tendremos las tablas con los datos y las claves primarias en la BD EMPRESA_<apellido>.
5. Comprueba que cada tabla tiene su clave primaria utilizando el comando `SHOW CREATE TABLE`.
6. Utilizando lo que respondiste en cuanto a borrados en cascada (en la práctica 3), añade las foreign key correspondientes mediante comandos `ALTER TABLE`.
7. Borra el empleado Franklin Wong. Si no puedes, indica el por qué y utiliza todos los borrados que sean necesarios hasta conseguirlo. Indica qué otros datos se han borrado automáticamente de la BD y por qué.
8. Realiza una copia de seguridad de la BD EMPRESA_<apellido>.

PRÁCTICA P7

El SGBD MySQL (3):

Consultas en SQL y funciones de fecha y hora con MySQL

OBJETIVOS

Para realizar esta práctica se necesita:

- No hay actividades previas.

Al terminar esta práctica se debe saber:

- Como crear una BD en la que las tablas soporten la integridad referencial.
- Cómo traer un fichero de texto con datos para una tabla de MySQL
- Cómo traer tablas de MSACCESS utilizando ODBC
- El lenguaje LMD de SQL: Instrucción SELECT y sus cláusulas:
 - FROM, WHERE
 - INNER JOIN, NATURAL JOIN
 - Funciones agregadas y atributos de agregación GROUP BY, HAVING
- Funciones básicas de Fecha y hora con MySQL.

<p>NOTA: Toda la ayuda adicional sobre el manejo de MySQL necesaria para la realización de esta práctica está en la práctica P5 y en el tema 4 de teoría.</p>
--

SINTAXIS DE SQL - LMD (Lenguaje de Manipulación de Datos) CONSULTAS

Realizar consultas (SELECT)

```
SELECT [DISTINCT | DISTINCTROW | ALL] select_expression,...  
[FROM table_references  
[WHERE where_definition]  
[GROUP BY {unsigned_integer | col_name | formula} [ASC | DESC], ...]  
[HAVING where_definition]  
[ORDER BY {unsigned_integer | col_name | formula} [ASC | DESC] ,... ]
```

Donde `select_expression` es un nombre de columna, fórmula o función agregada
y `where_definition` es una condición o combinación de condiciones mediante operadores lógicos AND y OR.

Reuniones entre tablas (JOIN)

Una reunión va en la cláusula FROM de una SELECT y puede corresponderse con uno de estos tipos:

- a) table_ref, table_ref
- b) table_ref [CROSS] JOIN table_ref
- c) table_ref INNER JOIN table_ref join_condition
- d) table_ref NATURAL [LEFT [OUTER]] JOIN table_ref
- e) table_ref NATURAL [RIGHT [OUTER]] JOIN table_ref
- f) table_ref LEFT [OUTER] JOIN table_ref join_condition
- g) table_ref RIGHT [OUTER] JOIN table_ref join_condition

Donde `table_ref` se define como:

table_name [[AS] alias]

y `join_condition` se define como:

ON conditional_expr | USING (column_list)

Importante:

Es posible que al realizar la reunión natural – NATURAL JOIN – si las tablas a reunir están separadas, se ejecute un PRODUCTO CARTESIANO, con lo que los resultados obtenidos serán erróneos. Por tanto, las tablas que tienen los campos en común han de aparecer seguidas en las reuniones naturales que impliquen más de dos tablas.

Funciones agregadas (condicionadas por la cláusula GROUP BY)

COUNT(expr)

Devuelve la cuenta del número de valores no NULL de las filas recuperadas por la SELECT.

COUNT(*)

Devuelve el contador del número de filas recuperadas aunque tengan valores NULL.

COUNT(DISTINCT expr,[expr...])

Devuelve el contador del número de valores no NULL diferentes.

```
mysql> SELECT COUNT(DISTINCT especialidad) FROM alumno;
```

AVG(expr)

Devuelve la media de expr

```
mysql> SELECT nombre_alumno, AVG(nota)
-> FROM alumno
-> GROUP BY nombre_alumno;
```

MIN(expr)

MAX(expr)

Devuelve el mínimo o máximo de expr. El atributo expr puede ser un string.

SUM(expr)

Devuelve la suma de expr. Si el conjunto a sumar no tuviera filas devuelve NULL.

FUNCIONES INCORPORADAS A SQL EN MYSQL – FUNCIONES DE FECHA Y HORA

Funciones básicas de fecha y hora

Fecha actual en formato YYYY-MM-DD	curdate() ó CURRENT_DATE
Hora actual en formato HH:MM:SS	curtime() ó CURRENT_TIME
Fecha y hora actual en formato YYYY-MM-DD HH:MM:SS	now() ó sysdate() ó CURRENT_TIMESTAMP

NOTA: Si se quiere que en un campo de una tabla aparezca por defecto la fecha y hora actual, al crear la tabla se pone ese campo como tipo TIMESTAMP.

Formato de una fecha

En MySQL a las fechas se les pueden dar distintos formatos con la función DATE_FORMAT utilizando distintos códigos. Existen más códigos que los que aparecen en los ejemplos, como son: %m, %M, %w, %W (disponibles por ejemplo en http://www.w3schools.com/sql/func_date_format.asp). También existe la función Format para dar formato a números en general.

En Access solo existe la función Format, que da formato tanto a números como a fechas.

Extracción de partes de una fecha:

Año	YEAR(fecha)
Mes	MONTH(fecha)
Día del mes	DAYOFMONTH(fecha) ó DAY(fecha)
Día de la semana en número	DAYOFWEEK(fecha) con 1 el domingo, 2 el lunes,..., 7 el sábado WEEKDAY(fecha) con 0 el lunes, 1 el martes,..., 6 el domingo
Día de la semana en letra	DAYNAME(fecha)

donde fecha es cualquier dato de tipo fecha (DATE, DATETIME ó TIMESTAMP)

Extracción de partes de una hora:

Hora	HOUR(hora)
Minutos	MINUTE(hora)
Segundos	SECOND(hora)

Añadir o quitar un intervalo fijo de tiempo a una fecha/hora:

DATE_ADD(fecha, INTERVAL incremento tipo)

Donde:

El incremento puede ser + para añadir ó - para quitar

El tipo puede ser YEAR, MONTH, DAY, HOUR, MINUTE ó SECOND.

Días transcurridos entre dos fechas:

DATEDIFF(fechaMasModerna, fechaMasAntigua)

Para calcular los días que hay entre dos fechas, mediante una resta de fechas, se puede usar la función que calcula el **Nº de días transcurridos desde el año 0:**

TO_DAYS(fecha)

PROBAR EN MYSQL LAS SIGUIENTES FUNCIONES QUE TRABAJAN CON FECHAS Y HORAS

Los dos guiones al principio de la línea (--) indican qué hace lo que aparece en los siguientes comandos.

-- FECHA ACTUAL

```
select current_date;
```

```
select curdate();
```

-- HORA ACTUAL

```
select current_time;
```

```
select curtime();
```

-- TODO JUNTO

```
select current_timestamp;
```

```
select now();
```

```
select sysdate();
```

-- DIAS TRANSCURRIDOS ENTRE DOS FECHAS COMO RESTA DE FECHAS

-- EJEMPLO ENTRE EL 10 Y EL 18 DE MAYO DE 2015

```
select to_days('2015-05-10');
```

```
select to_days('2015-05-18');
```

```
select to_days('2015-05-18')-to_days('2015-05-10');
```

-- EXTRACCION DE PARTES DE LA FECHA Y HORA ACTUAL

```
select now();
```

```
select year(now());
```

```
select month(now());
```

```
select dayofmonth(now());
```

```
select dayofweek(now());
```

```
select weekday(now());
```

```
select dayname(now());
select hour(now());
select minute(now());
select second(now());
```

```
-- AÑADIR/QUITAR UN INTERVALO DE TIEMPO
-- AÑADIR UN AÑO A LA FECHA DE HOY
select date_add(now(),interval 1 year);
-- QUITARLE CINCO MESES A LA FECHA DE HOY
select date_add(now(), interval -5 month);
```

- Las siguientes consultas son las que aparecen al final del TEMA 4 DE LA TEORÍA:

```
-- FECHA Y HORA ACTUAL
```

```
select now() as Fecha_Hora,
curdate() as SoloFecha,
curtime() as SoloHora;
```

```
-- PARTES DE UNA FECHA U HORA
```

```
select '2016-3-2 12:58:07' as Fecha_Hora,
year('2016-3-2 12:58:07') as Anio,
month('2016-3-2 12:58:07') as Mes,
day('2016-3-2 12:58:07') as Dia,
hour('2016-3-2 12:58:07') as Hora,
minute('2016-3-2 12:58:07') as Minuto,
second('2016-3-2 12:58:07') as Segundo;
```

```
-- DIA DEL MES tanto en número como en letra
```

```
select '2016-9-13 12:58:07' as Fecha_Hora,
month('2016-9-13 12:58:07') as NumeroMes,
monthName('2016-9-13 12:58:07') as NombreMes,
date_Format('2016-9-13 12:42:59','%b") as NombreMesAbreviado;
```

```
-- DIA DE LA SEMANA tanto en número como en letra
```

```
select '2016-9-13 12:58:07' as Fecha_Hora,
weekday('2016-09-13 12:42:59') as NumeroDiaSemana,
dayName('2016-09-13 12:42:59') as NombreDiaSemana,
date_Format('2016-09-13','%a") as NombreDiaSemanaAbreviado;
```

```
-- Ejemplo DateDiff: DIFERENCIA ENTRE DOS FECHAS
```

```
select '2016-04-05 00:00:00' as Fecha1,
'2016-04-13 12:42:59' as Fecha2,
datediff('2016-04-13 12:42:59','2016-04-05 00:00:00') as DiasEntreLasDosFechas;
```

```
-- EJEMPLO DATE_ADD
```

```
select '2016-7-22 20:35:44' as Fecha_Hora,
date_add('2016-7-22 20:35:44', interval 3 year) as Mas3Anios,
date_add('2016-7-22 20:35:44', interval -5 month) as Menos5Meses,
date_add('2016-7-22 20:35:44', interval 16 day) as Mas16Dias;
```

A la hora de realizar los siguientes apartados, si hace falta, **añade en las tablas los datos necesarios para que en todas las consultas se obtenga algún resultado.**

Si alguna de las siguientes bases de datos a consultar no existe, se deberá lanzar el script correspondiente, descargándolo desde el eGela de la asignatura “Bases de Datos”.

1. Descarga del eGela el fichero **scriptsCREACIONbds2020.zip** y descomprímelo.
2. Ejecuta el script **scripts_BDs_2020.sql** desde MySQL, con el comando SOURCE. Con esto se crearán las bases de datos EMPRESA, UNIVERSIDAD y BUQUES con los datos que aparecen en las páginas finales de estos enunciados.
3. Ejecuta el script **CREABDBIBLIOTECA2020.SQL**, con el comando SOURCE. Con esto se creará la base de datos BIBLIOTECA con los datos que aparecen en las páginas finales de estos enunciados.

Para cada consulta se debe entregar el comando SQL y el resultado que se obtiene.

SOBRE LA BD EMPRESA

Para poder entregar las consultas hechas, usa el comando TEE de MySQL antes de empezar:
TEE C:\TEMP\CONSULTAS_EMPRESA_APELLIDO.TXT

Realizar las consultas que aparecen desde la página 96 hasta la 98 en los apuntes de teoría del tema 4.

SOBRE LA BD UNIVERSIDAD

Para poder entregar las consultas hechas, usa el comando TEE de MySQL antes de empezar:
TEE C:\TEMP\CONSULTAS_UNIVERSIDAD_APELLIDO.TXT

- Crear en MySQL una BD llamada UNI_<mi primer apellido>
- Traer las tablas y datos de la BD UNIVERSIDAD_<mi primer apellido> de ACCESS utilizando ODBC.
- Poner las claves primarias a dichas tablas utilizando ALTER TABLE.
- **HACER SOLO ESTE APARTADO.** Realizar las consultas indicadas en la página 99 de los apuntes de teoría del tema 4.

➤ SOBRE LA BD BIBLIOTECA

- Para poder entregar las consultas hechas, pon el comando TEE de MySQL antes de empezar:
- TEE C:\TEMP\CONSULTAS_BIBLIOTECA_APELLIDO.TXT
- Realizar las consultas que aparecen desde la página 100 hasta la 106 en los apuntes de teoría del tema 4.

SOBRE LA BD BUQUES

Para poder entregar las consultas hechas, usa el comando TEE de MySQL antes de empezar:
TEE C:\TEMP\CONSULTAS_BUQUES_APELLIDO.TXT

- Realizar las consultas de las páginas 107 y 108 del tema 4 de teoría.

PRÁCTICA P8

PROYECTO VIDEOCLUB (1):

Enunciado y tareas del proyecto

ACTIVIDADES PREVIAS Y OBJETIVOS

Para realizar esta práctica se necesita:

- No hay actividades previas.

Al terminar el proyecto se debe saber:

- Los elementos básicos que se necesitan para implementar una aplicación que utiliza una base de datos: SGBD, conector y cadena de conexión, IDE (entorno de desarrollo) y controles que trabajan con datos.
- Asociar una base de datos a una aplicación.
- Insertar instrucciones SQL, con y sin parámetros, así como transacciones en una aplicación que accede concurrentemente a datos almacenados en una BD.

<p>NOTA: El material necesario para la realización de las prácticas correspondientes al proyecto está en los apuntes (en fotocopias) y en el egela de BD (material adicional, links y vídeos).</p>

TAREAS DEL PROYECTO

TAREA 1 (En grupos de dos personas):

1. Realizar el esquema entidad-relación de la base de datos de la aplicación del video club e implementar la base de datos diseñada, en MS ACCESS. Ampliar el diseño anterior y cargar los datos para crear **una base de datos conjunta para el proyecto del video club y para un sistema de recomendación de películas**. Toda la información para realizar estas tareas está en siguientes páginas de esta práctica.
2. Implementar – en VB.NET y ASP.NET – la funcionalidad que falta en la aplicación (tres páginas Web: Funciones ADMINISTRADOR, Funciones SOCIO y Catálogo PELÍCULAS). No es tarea de esta práctica ni diseñar las páginas, ni gestionar las cuentas Web de los usuarios (ni nombres de usuarios ni sus contraseñas), ya está hecho en una aplicación Web Inicial que se os facilitará. También está hecha el alta de un socio (para explicar el código en las clases de BD).

TAREA 2 (Individual):

Proponer una SELECT compleja e implementarla tanto en MS Access como en la aplicación.

En esta misma práctica y en las siguientes, se detalla el **enunciado** con las funcionalidades **para la aplicación** del video club, las características que ha de tener la **base de datos conjunta** que utilizará la aplicación, y **el resto del trabajo en ordenador** que implican estas dos tareas.

ENUNCIADO DEL PROYECTO

La práctica consiste en implementar la parte correspondiente a la base de datos para una aplicación Web de alquiler de películas de un video club que utiliza una base de datos (MS Access ó MySQL). En esta aplicación ya está implementado el login para entrar junto con la navegación por las distintas páginas de la web. Además, la primera vez que se entra el usuario registrado es dado de alta (activado) en el sistema de forma automática.

DATOS BÁSICOS SOBRE LOS SOCIOS Y LAS PELÍCULAS DEL VIDEO CLUB:

Por cada **socio del video club** la aplicación debe almacenar como mínimo su Nombre, Apellido y dirección, los datos de su cuenta para entrar a la aplicación – es decir, un nombre para el login del socio (único y similar a lo que utilizamos para entrar a eGela, como el “jlopez003”) y una contraseña para entrar –, crédito disponible en euros para alquilar películas y el momento en el que se le dio el alta en el video club. Un socio puede estar activo o haber sido dado de baja por el administrador del video club.

Por cada **película del video club** la aplicación debe almacenar al menos su código (único), título – nombre de la película (único) –, precio de alquiler y estado actual (alquilada, disponible o descatalogada).

Se debe saber en todo momento qué películas están alquiladas y disponibles, así como quién ha alquilado históricamente cada película y entre qué fechas aunque ya la haya devuelto.

FUNCIONALIDAD DE LA APLICACIÓN:

El sistema permite gestionar tanto las películas como los socios del video club, permite a cualquier usuario activo ver, alquilar o devolver películas y a cualquier persona que acceda a la web ver el catálogo de películas. Concretamente, las tareas son:

Alta de un Socio (Activar un usuario): tarea administrativa realizada automáticamente por el sistema cada vez que un nuevo usuario se registra. Cada vez que se da de alta a un socio, se almacena automáticamente el nombre de usuario que utilizó al registrarse en el sistema. El crédito estará siempre a 0 inicialmente. También se guardará automáticamente la fecha y hora a la que se registró (fecha y hora del alta). Los datos personales del usuario no se almacenan en este momento, sino que el propio socio los introduce posteriormente en la aplicación (mediante la funcionalidad modificar datos personales).

Cambiar el estado de un Socio: tarea administrativa que cambia el estado de un usuario registrado en el video club del estado activo al estado dado de baja o viceversa. Aunque la aplicación da de alta al usuario registrado de manera automática, el administrador del video club puede darle de baja si por ejemplo no devuelve una cierta cantidad de películas o hace mucho tiempo que no alquila nada. Para cambiar el estado de un socio basta con indicar su nombre de usuario registrado (el del login).

Alta de una Película: tarea administrativa que permite incluir una nueva película en el catálogo del video club. Cada vez que se da de alta a una película se almacenan todos sus datos. Su estado inicialmente será siempre disponible.

Baja de una película: tarea administrativa que elimina una película del catálogo del video club. Para dar de baja una película basta con indicar su título o su código (a elegir, lo que prefiera el programador). La aplicación debe asegurarse de que no está alquilada y avisar si es así. Si no está alquilada se debe eliminar todo rastro de esa película de la base de datos, si está alquilada se marca como descatalogada y a partir de este momento ya no se permite a nadie alquilarla más.

Modificar datos personales: tarea que permite al usuario activo modificar su dirección, nombre y apellidos y el resto de datos personales que tengamos almacenados sobre él. Nunca podrá modificar los datos de su cuenta ni su fecha de alta en el video club.

Aumentar crédito en euros: tarea que permite al usuario activo ingresar más dinero para alquilar películas.

Alquilar y/o devolver una película: tarea que permite al usuario activo gestionar el alquiler o devolución de las películas que alquiló.

Ver catálogo y estrenos: cualquier usuario que entre en la página principal de la Web aunque no esté registrado puede ver todas las películas del video club, si están disponibles o no y todos los estrenos, considerándose películas de estreno las adquiridas por el video club en la última semana.

En resumen, las funciones que se desea informatizar se dividen en tres tipos: tareas administrativas del video club, tareas propias de los usuarios activos y tareas para cualquier usuario de la web. De este modo, la aplicación podrá ser utilizada por **TRES TIPOS DISTINTOS DE USUARIOS:**

- **Administrador del video club:** puede realizar cualquiera de las funciones de la aplicación. Es el único que puede encargarse de las tareas administrativas. Se almacena como un socio más.
- **Socio:** usuario registrado en el sistema y dado de alta (activo) en el video club. No puede realizar las tareas administrativas del video club, pero sí todas las demás.
- **Usuario:** cualquiera que entre en la Web. No puede realizar ni las funciones administrativas ni las de los socios.

Existe un único usuario que actúa como administrador del video club, es decir, se encarga de las altas y bajas de los socios y las películas y ya está registrado en el sistema como **administrador** con contraseña **123456**. El resto de usuarios que se han registrado y están dados de alta forman el conjunto de socios del Video Club, que actúan a nivel de sesión y que entran a la aplicación haciendo login en el sistema utilizando su nombre de usuario y contraseña (inventadas ambas al registrarse).

DATOS PARA EL SISTEMA DE RECOMENDACIÓN DE PELÍCULAS

Los Sistemas de Recomendación intentan personalizar al máximo lo que ofrecerán a cada usuario. Esto es posible por la cantidad de información individual que podemos recabar de las personas y nos da la posibilidad de mejorar la experiencia del internauta sin ofrecer productos a ciegas. Un sistema de recomendación es un algoritmo que aprovecha la información de apps y webs para predecir los siguientes ítems (productos, canciones, películas, etc.) que querrá consumir un usuario en particular. Algunos de los casos más conocidos de uso de esta tecnología son *Netflix* recomendando series y películas, *Spotify* sugiriendo canciones y artistas, o *Amazon* ofreciendo productos de venta cruzada tentadores para cada usuario. Otros ejemplos son *Google* que nos sugiere búsquedas relacionadas, *Android* aplicaciones en su tienda y *Facebook* amistades. O las típicas *lecturas relacionadas* en blogs y periódicos.

Para poder implementar un sistema de recomendación necesitaremos, los “ítems” y las valoraciones de los usuarios. Los ítems pueden ser, canciones, películas, productos, o lo que sea que queremos recomendar. Entonces nos quedará una matriz, donde la intersección entre fila y columna es una valoración o rating del usuario. El algoritmo del sistema rellenará los huecos para recomendar ítems al usuario. Hay más información en <https://www.aprendemachinelearning.com/sistemas-de-recomendacion/>

Así, en nuestro caso, para poder implementar un sistema de recomendación de películas necesitamos guardar en la base de datos, los datos de los usuarios/internautas y de las películas. Y cada combinación (usuario, película) contendrá la valoración hecha por ese usuario para esa película. Evidentemente, ni un usuario ha visto todas las películas, ni una película ha sido vista por todos los usuarios. De forma adicional, guardaremos también todas las etiquetas (tags) puestas a las películas, aunque estén repetidas.

Se trata de diseñar, crear y cargar una base de datos que contenga:

- toda la información necesaria para implementar las funcionalidades del Video Club y
- todos los datos necesarios para poder implementar (en otras aplicaciones distintas a este proyecto) un sistema de recomendación de películas.

Todo el material para realizar esta tarea está disponible en el eGela de Bases de Datos, en la carpeta “ENUNCIADOS CLASES PRÁCTICAS Y FICHEROS NECESARIOS PARA REALIZARLAS”.

Los datos contenidos en **Datos_Originales_Todas_Peliculas.accdb** servirán para cargar los datos de las películas de 2018 en la BD a crear, y han sido extraídos directamente del fichero ml-latest-small.zip descargado de <http://grouplens.org/datasets/movielens/>. Los datos contenidos en los ficheros de texto (.csv), **movie-titles.csv**, **movie-ratings.csv** y **movie-tags.csv** corresponden a películas de años anteriores, y han sido proporcionados por el equipo docente de la asignatura de “Ingeniería del Software”. Estos ficheros csv, junto con la explicación de lo que contienen, están en el fichero comprimido **DatosEuskoFlyx.zip**.

Pasos a dar para el diseño, implementación y carga inicial de la base de datos del VideoClub (VC) con datos para Sistema de Recomendación de películas (SR).

1. **Crear el esquema E/R** y las tablas (con sus claves y relaciones) para la base de datos del videoclub, según lo indicado en las páginas anteriores de esta práctica **P8**.
2. **Ampliar el esquema E/R** del videoclub, añadiéndole nuevas entidades, atributos y vínculos que reflejen la información correspondiente al sistema de recomendación de películas. Basándose en la información contenida en **Datos_Originales_Todas_Peliculas.accdb**, los datos necesarios son:
 - a. Los códigos de películas y títulos con su año de estreno entre paréntesis. Campos de la tabla **películas_UTF8**: movieId y title.
 - b. Las valoraciones de películas por parte de los usuarios. Campos de la tabla **valoraciones**: userId, movieId y rating. El usuario que aparece en estos datos no tiene nada que ver con el socio del Video Club. Un usuario nunca valora más de una vez una determinada película.
 - c. Las etiquetas de películas puestas por los usuarios. Campos de la tabla **etiquetas**: movieId y tag. Las combinaciones (IdPelícula, etiqueta) pueden repetirse. *Ayuda*: Una posible solución para una entidad/tabla cuyos atributos no pueden formar una clave es ponerle directamente una clave autonumérica. Existen también otras soluciones, pero ésta es la más sencilla.
3. **Crear una base de datos** en MS Access llamada **Videoclub_Apellido**, con el diseño decidido en los pasos anteriores.

Crea la tabla SOCIO, con los siguientes campos como mínimo:

- usuarioLogin → cadena de máximo 50 caracteres. Clave primaria.
- nombre_apellido → cadena de máximo 50 caracteres. Valor por defecto: Introduzca nombre y apellidos.
- Direccion → cadena de máximo 100 caracteres. Valor por defecto: Introduzca dirección.
- Credito → número real doble con dos decimales. Valor por defecto: 0
- Fecha_Hora_Alta → dato de tipo fecha y hora (TIMESTAMP en SQL). Valor por defecto: la fecha y hora actual.
- Estado → un solo carácter. Valor por defecto: A

Añade los datos del administrador del videoclub en la tabla SOCIO:

- usuarioLogin: administrador.
- nombre_apellido: tu nombre y tu primer apellido.
- dirección, crédito y fecha_hora_alta: lo que se quiera
- estado: A

Crea las demás tablas de tu esquema, poniendo las claves de las tablas y las restricciones de integridad.

4. Cargar los datos iniciales en la base de datos anterior, utilizando la importación / exportación de datos, desde los ficheros que se indican y que están colgados en eGela:
 - a. Importar las **películas inicialmente disponibles en el videoclub**, que serán las que se han estrenado en el 2018 y que están ya separadas en el fichero de texto *películas2018.csv* (41 películas). El contenido de este fichero se debe importar a la tabla PELICULA, de la BD del Video Club. El estado inicial de estas películas ha de ser '*Disponible*' y su precio 3 euros.
 - b. Cargar **los datos de las valoraciones y etiquetas correspondientes a las películas de 2018**. Esta información se extraerá del fichero *Datos_Originales_Todas_Peliculas.acddb*. Para ello se harán dos consultas en esta base de datos (una para las valoraciones y otra para las etiquetas) y posteriormente se importarán los datos. Hay 91 valoraciones y 26 etiquetas que corresponden a las películas de 2018.
 - c. Cargar los **datos correspondientes a las películas exclusivas del sistema de recomendación**, que son las anteriores a 2018, importando directamente los datos desde los ficheros de texto *movie-titles.csv* (100 películas), *movie-ratings.csv* (338355 valoraciones) y *movie-tags.csv* (41980 etiquetas). Estos 3 ficheros se encuentran en el fichero comprimido **DatosEuskoFlyx.zip**, colgado en eGela. El estado inicial de todas estas películas ha de ser '*Descatalogada - SR*', lo que indicará que esa película no está disponible en el videoclub, sólo en el SR.
5. Separar los datos del Sistema de Recomendación, creando una consulta/vista que solo tenga la tabla de películas exclusivas del SR (es decir, solo los códigos y títulos de las películas que había en *movie-titles.csv*).

PRÁCTICA P9

PROYECTO VIDEOCLUB (2):

Aplicación Web ASP.NET con VB.NET y MS ACCESS

(Tarea 1 del proyecto – En grupo o individual)

ACTIVIDADES PREVIAS Y OBJETIVOS

Para realizar esta práctica se necesita:

- Formar el grupo, si se va a hacer el trabajo en parejas.
- El diseño relacional de la BD del proyecto del grupo. Este diseño contiene las tablas (campos y claves) y las relaciones (con integridad referencial) correspondientes al modelo E/R de la base de datos del proyecto del grupo.

Al terminar esta práctica se debe saber:

- Cómo conectar una base de datos con una aplicación.
- Los elementos básicos del entorno (Visual Studio) en las aplicaciones Web ASP.NET: Solución y Proyecto, opción Ver, Site.Master, tipos de ficheros .aspx y .aspx.vb, login y ASPNETXXX.mdf, etc.
- Cómo comenzar el proyecto del VideoClub Web.

<p>NOTA: El material necesario para la realización de las prácticas correspondientes al proyecto está en los apuntes (en fotocopias) y en el egela de BD (material adicional, links y vídeos).</p>

Programas y ficheros necesarios para tu ordenador

1. Microsoft ACCESS 2007 o superior (y su conector OLEDB si no está).
2. Microsoft VISUAL STUDIO COMMUNITY (version gratuita). Web de descarga <https://www.visualstudio.com/es/downloads/>
 - a. Instalar como mínimo la parte: “Desarrollo de ASP.NET y Web”
 - b. Para poder usar el programa indefinidamente, crear una cuenta personal en Microsoft.
3. Fichero **ApliWebVideoClubINIAccess2017_18.zip** (disponible en el eGela de BD). Contiene la aplicación Web inicial (para usar una base de datos implementada en MS Access).
4. Para una mejor comprensión de estas prácticas. se recomienda que CONFIGURES TU ORDENADOR, de forma que se vean las extensiones de los ficheros. Pasos en Windows 10: (1) Abrir cualquier carpeta (2) pinchar en “Vista” (3) Marcar “Extensiones de nombre de archivo”

Pasos para probar la aplicación:

- PASO 1. Crear la BD en MS ACCESS del grupo (por lo menos la tabla SOCIO y el administrador del video club tal y como se indica en la práctica P8).
- PASO 2: Abrir el proyecto, conectar la BD y ejecutar la aplicación Web.

IMPORTANTE para trabajar después con la aplicación

Para poder trabajar en el proyecto siempre se debe tener: (1) la última versión de la aplicación Web y (2) la última versión de la BD.

PASO 1. CREAR LA BD DEL VIDEOCLUB DEL GRUPO

En MS ACCESS:

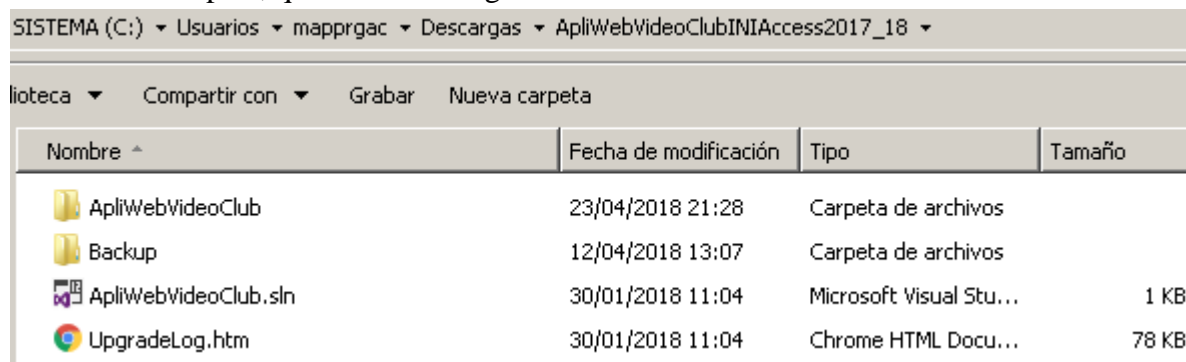
1. Abrir y corregir si es necesario la BD individual que esté mejor.
2. Guardarla como “Base de Datos de Microsoft Access (2002-2003)” con el nombre *videoclub_Apellido1_Y_Apellido2.mdb* donde Apellido1 y Apellido2 son los primeros apellidos de los dos componentes del grupo en orden alfabético. No poner eñes ni acentos en los apellidos.

PASO 2. ABRIR EL PROYECTO, CONECTAR LA BD A LA APLICACIÓN Y COMPROBAR QUE FUNCIONA

Lee las **PÁGINAS DE 1 A 4 DE LA TEORÍA DEL TEMA 5**. Después realiza los siguientes pasos por orden.

PASO 2.1. APERTURA DEL PROYECTO

1. Descarga y visualiza el vídeo *VideoClub_VIDEO1.avi* del eGela de BD.
2. Descomprime el fichero **ApliWebVideoClubINIAccess2017_18.zip** de la aplicación inicial y mira el contenido de la carpeta, que es el de la figura.



Nombre	Fecha de modificación	Tipo	Tamaño
ApliWebVideoClub	23/04/2018 21:28	Carpeta de archivos	
Backup	12/04/2018 13:07	Carpeta de archivos	
ApliWebVideoClub.sln	30/01/2018 11:04	Microsoft Visual Stu...	1 KB
UpgradeLog.htm	30/01/2018 11:04	Chrome HTML Docu...	78 KB

En Visual Studio, todos los elementos necesarios de una aplicación se guardan en una carpeta, llamada carpeta de la solución, que en este caso es *ApliWebVideoClubINIAccess2017_18*. Contesta a las siguientes preguntas:

- a. La carpeta *ApliWebVideoClub* es la carpeta del proyecto. Ábrela. Aparte de otras carpetas, ¿Qué tipos de ficheros (extensiones) hay en ella? ¿Qué nombres de ficheros (sin extensiones)? ¿Qué relación hay entre los ficheros con extensión *.aspx* y *.aspx.vb*? ¿Qué hay en las carpetas *Account* y *App_Data*?
 - b. El fichero de la solución (*.sln*), ¿Para qué sirve?
3. Siendo *Apellido1* y *Apellido2* los primeros apellidos de los dos componentes del grupo en orden alfabético, cambia el nombre de la carpeta de la solución por *ApliWebVideoClub_Apellido1_Y_Apellido2*.
 4. Abre el proyecto.
 5. En Visual Studio, ¿Qué hay en la opción del menú *Ver*?

PASO 2.2. CONEXIÓN DE LA BD CON LA APLICACIÓN

Indica la cadena de conexión a la base de datos *videoclub*. Para ello, busca y cambia la BD, que aparece resaltada en negrita, por tu BD.

```
Dim cadenaConexion As String = "Provider = Microsoft.ACE.OLEDB.12.0 ; Data Source =  
C:\TEMP\VideoClub.accdb"
```

Inicialmente, esta cadena está en el código VB.NET de *Default* y de *FormularioFuncionesSocio*.

PASO 2.3. EJECUCIÓN DE LA APLICACIÓN WEB

1. Descarga y visualiza el vídeo *VideoClub_VIDEO2.avi* del eGela de BD.
2. Ejecuta la aplicación Web desde la página maestra (*Site.Master*) e *Inicia Sesión* como usuario administrador con contraseña 123456. Comprueba que puede entrar en las tres opciones/páginas de la aplicación: *Funciones ADMINISTRADOR*, *Funciones SOCIO* y *Catálogo PELICULAS*.

3. Cierra la sesión e Inicia sesión como un socio nuevo pinchando en “Registrarse”. Comprueba que en la BD videoclub aparezca una nueva fila en la tabla SOCIO con los datos de la cuenta creada.

Si al probar la aplicación sale el error: **"El proveedor 'Microsoft.ACE.OLEDB.12.0' No está registrado en el equipo local"**, es que no tienes el conector OLEDB necesario. Sigue los siguientes pasos:

- Cierra MS Visual Studio si está abierto
- Ve al enlace <https://www.microsoft.com/es-es/download/details.aspx?id=13255> y pincha en Descargar
- Descarga el fichero "AccessDatabaseEngine.exe" (NO descargues el de 64)
- Ejecútalo
- Vuelve a probar la aplicación.

La siguiente figura muestra la página que se visualiza en el navegador al entrar en las funciones de administrador:

MI APLICACIÓN ASP.NET Pantalla de bienvenida **administrador!** [[Cerrar sesión](#)]

[Página principal](#) [Acerca de](#) [Funciones ADMINISTRADOR](#) [Funciones SOCIO](#) [Catálogo PELICULAS](#)

FUNCIONES ADMINISTRADOR

Realice todas las funciones que quiera . Rellene los datos oportunos y pulse el botón.

Cambiar estado de un socio

UsuarioLogin del socio a cambiar de estado

<Visualizar aquí los datos del socio>

Dar de alta una película

Código de la película a dar de alta

<Poner aquí el resto de datos a pedir>

Dar de baja una película

Código de la película a dar de baja

<Visualizar aquí los datos de la peli>

4. Fíjate en el código de la página maestra (Site.Master) en Visual Studio, ¿Qué elementos de la figura aparecen en ese código?
5. ¿En qué fichero están los demás elementos de la página que aparecen en la figura?
6. Registra un nuevo socio (con Iniciar Sesión) y comprueba que funciona entrando en “Funciones SOCIO” . Comprueba también que el nombre de este socio aparezca en la base de datos.

PRÁCTICA P10

PROYECTO VIDEOCLUB (3):

Controles de servidor ASP.NET para las páginas Web

(Tarea 1 del proyecto – En grupo o individual)

ACTIVIDADES PREVIAS Y OBJETIVOS

Para realizar esta práctica se necesita:

- La aplicación Web y la BD de tu grupo resultantes de la práctica anterior. En la BD en ACCESS deben existir al menos la tabla de socios y la de películas.

Al terminar esta práctica se debe saber:

- Utilizar los controles con enlace a datos ASP.NET.

NOTA: El material necesario para la realización de esta práctica está en los apuntes (en fotocopias) y en el egele de BD (material adicional, links y videos).

Antes de empezar, lee las **PÁGINAS DE 5 A 12 DEL TEMA 5 DE TEORÍA**.

CONTROLES DE SERVIDOR ASP.NET EN LA PÁGINA WEB

Las siguientes figuras muestran ejemplos de controles de servidor ASP.NET implementados en las Páginas Web de la aplicación VideoClub. Se puede ver su funcionamiento en los vídeos *VideoClub_FUNCIONES_ADM.avi* y *VideoClub_FUNCIONES_SOCIO.avi* colgados en el egela de BD.

MI APLICACIÓN ASP.NETPantalla de bienvenida **administrador!** [[Cerrar sesión](#)]

[Página principal](#) [Acerca de](#) [Funciones ADMINISTRADOR](#) [Funciones SOCIO](#) [Catálogo PELÍCULAS](#)

FUNCIONES SOCIO

Realice todas las funciones que quiera . Rellene los datos oportunos y pulse el botón.

Modificar mis datos personales
Nombre y apellidos
Dirección

Aumentar mi crédito
Introduzca aquí la cantidad en euros

Alquilar película
Seleccione de la lista la película a alquilar

Devolver película
Seleccione de la lista la película a devolver

CATÁLOGO PELÍCULAS

Título	Precio	Estado
500 días juntos	1	Descatalogada
Al filo del mañana	1	Alquilada
Aladdin	1	Disponible
Alien	2	Disponible
Atonement	1	Disponible
Disconnect	4	Disponible
El atlas de las nubes	5	Disponible
El club de la lucha	1	Descatalogada
El exorcismo	2	Disponible
El juego de Ender	2	Disponible

ESTRENOS SEMANA (fecha de hoy: 17/04/2016)

Título	Precio	FechaDisponible
Julieta	4	17/04/2016 13:05:40
El pregón	4	11/04/2016 18:46:33
Kiki el amor se hace	4	11/04/2016 18:46:00

FUNCIONES ADMINISTRADOR

Realice todas las funciones que quiera . Rellene los datos oportunos y pulse el botón.

Cambiar estado de un socio

UsuarioLogin del socio a cambiar de estado

DataList Nombre y apellido: Pepe Lopetegui
Dirección: Dr. Areilza 20, 3ºB
Crédito: 4567
Fecha y hora del alta: 04/04/2016 18:12:10
Estado: A

DropDownList

pepe
socio
pepe
juan

Cambiar estado

Mostrar datos socio

DISEÑO DE CONTROLES BÁSICOS DE LA PÁGINA WEB:

TextBox (caja de texto), Button (botón) y Label (etiqueta)

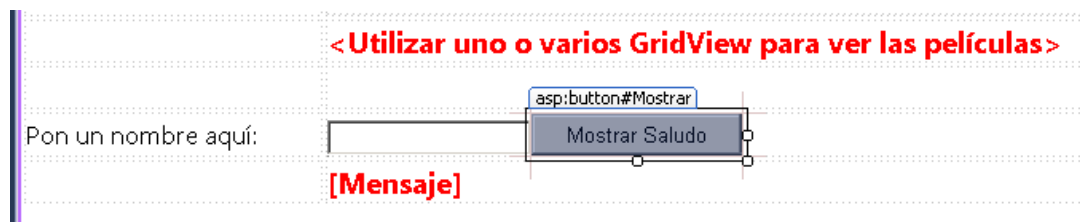
Ejemplo de diseño de varios controles básicos ASP.NET (TextBox, Button y Label) de la página Web y programación en VB.NET de uno de ellos.

Paso 1. Creación de un objeto de control.

Abre la vista “Diseño” de la página del catálogo de películas (FormVerPELICULAS.aspx). Fíjate que la parte blanca es una tabla. Crea tres *objetos de control básicos* en ella:

1. Pincha en cualquier parte de la fila **<Utilizar...>** y añade una nueva fila debajo (clic botón derecho) en el diseño de la página, para incluir en ella los nuevos controles.
2. Escribe en la columna de más a la izquierda: *Pon un nombre aquí:*
3. Desde el cuadro de herramientas, añade (arrastra al lugar donde quieres que vayan) los siguientes objetos de control:
 - a. Un Textbox cuyo ID sea *Nombre*,
 - b. Un Button cuyo ID sea *Mostrar* con el texto *Mostrar Saludo* y
 - c. Un Label cuyo ID sea *Mensaje*, sin texto y debajo de los anteriores en otra fila nueva.

Seleccionando el objeto de control en el diseño de la página Web, mediante clic en el botón derecho del ratón, se accede a sus propiedades igual que en Access. Ejemplos de propiedades de los controles: ID y Text.



Paso 2. Codificación de un objeto de control.

Programa el botón Mostrar.

1. Accede a su código VB.NET (haciendo Doble clic en el diseño del botón). Aparece la subrutina en la que se indican las instrucciones a ejecutar cuando se haga clic en el botón Mostrar:

```
Protected Sub Mostrar_Click(sender As Object, e As EventArgs) Handles Mostrar.Click
    ' <Aquí van las instrucciones para el clic en el botón Mostrar>
End Sub
```

2. Introduce dentro las siguientes instrucciones para el botón (lo verde son comentarios en VB.NET):

```
Dim varNombre As String 'Se declara una variable de tipo string llamada varNombre
varNombre = Me.Nombre.Text 'Se asigna a esa variable el valor de la caja de texto
Me.Mensaje.Text="Hola " & varNombre & " !" 'Se concatena con & y se pone en el Label
```

Paso 3. Prueba del funcionamiento del control.

Para probar el funcionamiento del botón, ejecuta la aplicación Web, introduce un nombre en la caja de texto y haz clic en el botón. Se tiene que ver el mensaje.

Pon un nombre aquí:

Pepe

Mostrar Saludo

Hola Pepe !

DISEÑO DE CONTROLES ENLAZADOS A DATOS:

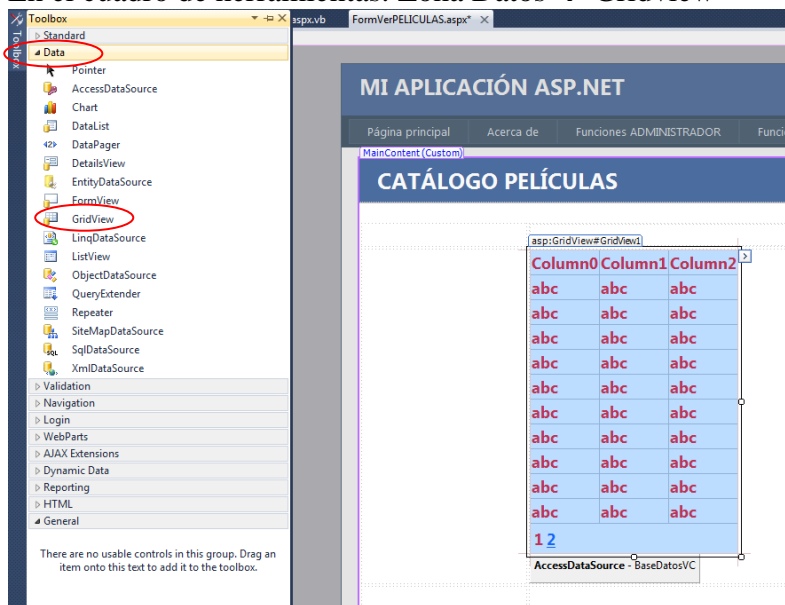
GridView, DataList y DropDownList

El siguiente ejemplo indica cómo se visualizan en un control GridView los datos provenientes de una base de datos, utilizando el asistente que proporciona Visual Studio para este control. Otros controles ASP.NET con enlace a datos son DataList y DropDownList, cuyos pasos de diseño son similares al que se muestra en este ejemplo para un GridView.

Paso 1. Creación del elemento de control con enlace a datos

Abre el diseño de la página Web “Catálogo Películas” (fichero .aspx). Sitúate con el ratón en **<Utiliza...>**, borra la frase y arrastra el elemento GridView del cuadro de herramientas (si no se ve el cuadro desplegar menú Ver del Visual Studio y seleccionarlo)

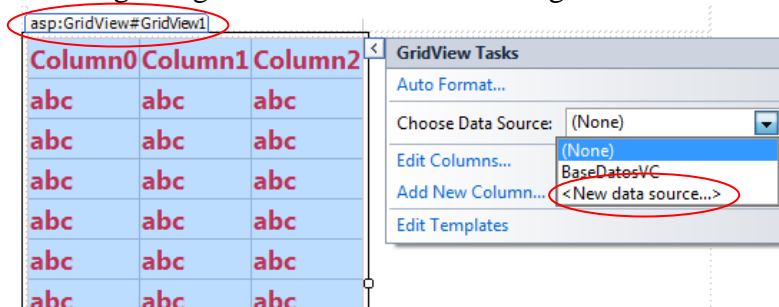
En el cuadro de herramientas: Zona Datos → Gridview



Pasos 2 y 3. Creación de un nuevo control de origen de datos para el elemento y configuración de ese origen de datos.

Elige el origen de los datos, es decir, la base de datos que contiene los datos a visualizar e indica qué es lo que se va a visualizar: tabla entera, una SELECT, etc...

En las tareas del GridView: elegir origen de datos: → <nuevo origen de datos...>



Se abre el asistente, que nos pide todos los datos necesarios tanto sobre la base de datos (para establecer la cadena de conexión) como sobre los datos concretos a visualizar. Concretamente, se debe:

- Elegir el tipo y la base de datos, en nuestro caso “Base de datos de Access” y el nombre de la BD con la ruta completa, por ejemplo C:\TEMP\videoclub.accdb. Puede que nos salga este error:

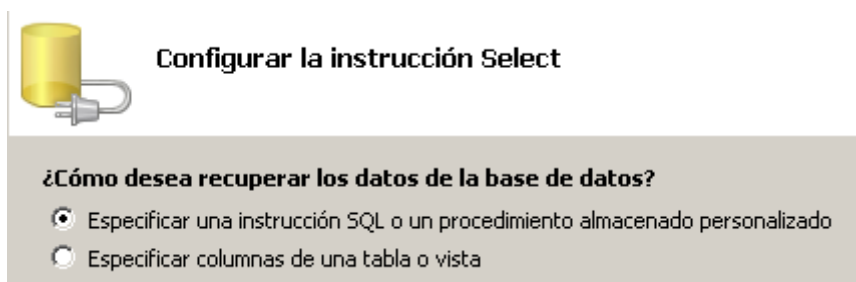


No se pudo recuperar el esquema de base de datos para esta conexión. Asegúrese de que la configuración de conexión es correcta y la base de datos está en línea.

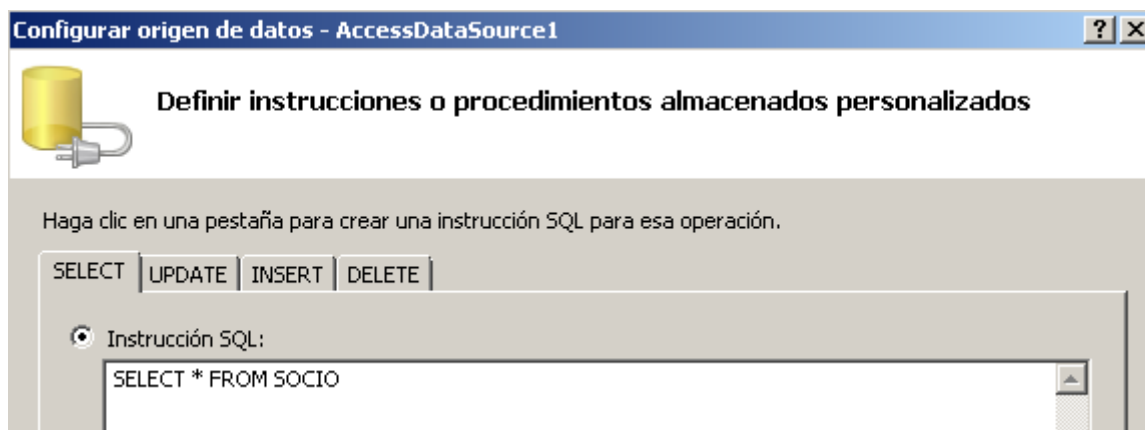
No se reconoce el formato de base de datos
'c:\TEMP\videoclub.accdb'.

Si este es el caso, una manera de arreglarlo es cambiar el formato de la base de datos a una versión anterior. ¿Cómo? Abrimos la BD con MS ACCESS y la guardamos como “Base de Datos de Access 2002-2003 (*.mdb)”. Ahora tendremos una copia con el mismo nombre que la BD inicial y extensión .mdb, que será la que tenemos que utilizar. También habrá que cambiar la BD por esta otra en la cadena de conexión de la aplicación Web (ver cómo en la práctica P9).

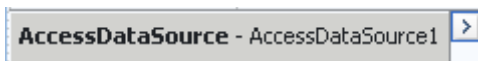
- Configurar la instrucción SELECT. Aparecen dos opciones: elegimos la primera “Especificar una instrucción SQL o un procedimiento almacenado personalizado”.



Al dar a “Siguiente”, sale la pantalla donde tecleamos la SELECT:



- En la siguiente pantalla podremos ver el resultado de la SELECT pulsando el botón “Consulta de prueba”. Si todo es correcto, damos a “Finalizar” para terminar.



Al terminar aparecerá algo similar a  justo debajo del GridView. En él se podrá cambiar la configuración inicial dada al origen de datos, si es necesario.

En este ejemplo se ha creado un origen de datos que visualiza todos los datos de la tabla SOCIO, es decir, se ha indicado la instrucción SQL:

`SELECT * FROM SOCIO;`

Haciendo clic en la pestaña **código** fuente de la página .aspx (Source de la figura) en vez de la pestaña diseño de las figuras anteriores, se ve el código en ASP.NET – que Visual Studio ha creado automáticamente – y que corresponde a los dos controles y a la SELECT asociada.

Objeto de Control de enlace a datos de tipo GridView

```

<asp:GridView
    ID="GridView1" runat="server" AllowPaging="True"
    AutoGenerateColumns="False" DataSourceID="BaseDatosVC"
    DataKeyNames="usuarioLogin">
    <Columns>
        <asp:BoundField DataField="usuarioLogin" HeaderText="usuarioLogin"
            ReadOnly="True" SortExpression="usuarioLogin" />
        <asp:BoundField DataField="nombre_apellido" HeaderText="nombre_apellido"
            SortExpression="nombre_apellido" />
        <asp:BoundField DataField="Direccion" HeaderText="Direccion"
            SortExpression="Direccion" />
        <asp:BoundField DataField="Credito" HeaderText="Credito"
            SortExpression="Credito" />
        <asp:BoundField DataField="Fecha_Hora_Alta" HeaderText="Fecha_Hora_Alta"
            SortExpression="Fecha_Hora_Alta" />
        <asp:BoundField DataField="Estado" HeaderText="Estado"
            SortExpression="Estado" />
    </Columns>
</asp:GridView>

<asp:AccessDataSource ID="BaseDatosVC" runat="server"
    DataFile="videoclub.accdb" SelectCommand="SELECT * FROM [SOCIO]" />

```

Objeto de Control de origen de datos de tipo AccessDataSource (asociado al GridView1)

Quando se execute esta página, en el navegador se verán los datos en forma de tabla básica con una apariencia similar a la de la figura.

usuarioLogin	nombre_apellido	Direccion	Credito	Fecha_Hora_Alta	Estado
administrador	Juan Pedro	Gran vía 22	0	06/04/2013 11:59:25	A
pepe115	Pepe Liz	La Concha 28	0	25/03/2015 13:16:01	A
juan1	Juan Garci	Indautxu 3	0	25/03/2015 13:16:04	A
maria34	Maria de la O	Unamuno 122	0	25/03/2015 13:16:08	A
koldo22	koldo Bru	Bidezabal 45	0	25/03/2015 13:16:13	B

Paso 4 (Opcional). Configuración del elemento de control con enlace a datos.

Con las opciones del control GridView (que aparecen seleccionando la flecha que hay en el control del paso 1) se le puede dar formato a nuestro gusto y añadirle otras funcionalidades que dependen de la configuración dada al origen de datos asociado a él.

DISEÑO DE LA PÁGINA WEB CATÁLOGO DE PELÍCULAS

1. De momento, NO BORRES el ejemplo de TextBox, Button y Label. **HAY QUE ENTREGARLO TAMBIÉN.**
2. Utilizando la solución del grupo, implementa la página correspondiente a la opción del menú del sitio Web *Catálogo PELÍCULAS*. La funcionalidad a implementar es la de “Ver catálogo y estrenos” (Los detalles están en el enunciado de la práctica P8). Para ello, sigue el diseño que aparece en el primer apartado de esta práctica (página 60). Para diseñar el GridView que aparece a la izquierda, puedes reutilizar el ejemplo que acabas de hacer.
3. Introduce a mano algunas películas en la BD del grupo, para que salga algún resultado, y prueba la página.

PRÁCTICA P11

PROYECTO VIDEOCLUB (4):

Consulta SELECT en MS ACCESS y en la aplicación

(Tarea 2 del proyecto – Individual)

ACTIVIDADES PREVIAS Y OBJETIVOS

Para realizar esta práctica se necesitan:

- La BD en ACCESS del videoclub de tu grupo.
- Tu propuesta de consulta (SELECT) basada en el diseño de la BD del videoclub de tu grupo, con las correcciones del profesor.

Al terminar esta práctica se debe saber:

- Introducir instrucciones SQL complejas tanto en MS Access como en la aplicación Web.

NOTA: El material necesario para la realización de esta práctica está en los apuntes (en fotocopias) y en el egela de BD (material adicional, links y videos).

Las tareas que se describen a continuación se tienen que realizar **DE MANERA INDIVIDUAL**.

1. TERMINAR SELECT PROPUESTA Y GUARDARLA EN LA BD

Consiste en completar tu consulta siguiendo las indicaciones del profesor, probarla en el SQL de MS Access y guardar la propia consulta resuelta en la BD (para entregar en la parte individual). Para ello debes añadir a la base de datos en MS Access del videoclub del grupo, todos los datos necesarios para que la consulta planteada ofrezca algún resultado al ejecutarse.

Detalle de las tareas a realizar:

1. Coge la BD Access del videoclub de tu grupo y la aplicación Web realizadas hasta este momento.
2. Si es necesario, añade al diseño de la base de datos todo lo necesario para que se pueda implementar tu consulta (tanto campos como nuevas tablas).
3. Completa tu consulta hasta que tenga todas las cláusulas que se piden.
Si la propuesta que se entregó por escrito no estaba completa se puede añadir lo que falte, pero respetando la idea original.
La SELECT a implementar debe contener todas las cláusulas vistas en SQL: JOIN, WHERE, GROUP BY, HAVING, ORDER BY y alguna función de FECHA/HORA. Si no se te ocurre una sola SELECT, puedes implementar varias SELECT con tal de que al final aparezcan todas las cláusulas descritas.
También debe tener uno o más parámetros, cuyo valor vendrá desde uno o varios TextBox.
4. Introduce en la BD los datos que sean necesarios para que se obtenga algún resultado. Los que puedas mediante la aplicación Web y los demás a mano.
5. Si te resulta más cómodo, puedes probar primero la consulta en ACCESS y directamente en la BD. Se prueba igual que Visual Studio, poniendo un '?' en cada sitio donde vaya un parámetro. Al ejecutarla nos irá pidiendo los valores concretos, que habrá que darle.
6. Crea una nueva página Web en la aplicación, tal y como se indica en la siguiente página de esta práctica.
7. Implementa la consulta en el GridView y pruébala hasta que funcione.
8. Guarda la consulta final en la aplicación Web.

Particularidades del lenguaje SQL en MS ACCESS:

Ante dudas de sintaxis en la SELECT ir a la Vista Diseño, consultar la Ayuda general de MS Access (icono '?' en la parte superior de la ventana del programa) o ir a Internet. Algunas particularidades conocidas:

- No existe el NATURAL JOIN: En vez de esto poner INNER JOIN...ON...
- No existe % para indicar varios caracteres en la cadena del LIKE: Usa * en vez de %
- No existe COUNT(*): En vez de esto poner COUNT(NombreDelCampoAContar)
- Los campos deben aparecer calificados, esto es, de la forma TABLA.Campo

Además, se debe tener en cuenta que algunos errores pueden ser provocados por un MAL DISEÑO de la BD.

2. CREAR UNA NUEVA PÁGINA EN LA APLICACIÓN WEB E INCLUIR TU SELECT

Consiste en crear una nueva página en la aplicación Web, en la que la página que visualiza la consulta aparezca como una opción más del Menú del sitio Web.

Detalle de las tareas a realizar:

1. Crea una nueva página en la aplicación Web que siga el mismo formato que la página maestra.
 - a. En la ventana del “Explorador de soluciones” selecciona el proyecto ApliWebVideoClub
 - b. Haz clic on el botón derecho, Agregar nuevo elemento → Web Forms → Formulario Web con página maestra
 - c. Abajo poner el nuevo nombre para la página. Ponerle como nuevo nombre: MiEstadistica.
2. Añadir una tabla en el diseño de esta nueva página Web, arrastrando desde el cuadro de herramientas los elementos necesarios
 - a. Los elementos están en la parte html del cuadro de herramientas
 - b. Arrastrar a MainContent un elemento div
 - c. Arrastrar a div un elemento table
3. Modifica el menú del sitio Web
Añade una nueva opción a las opciones del menú del sitio Web para que se pueda acceder desde este menú a la nueva página Web. En esta nueva opción debe visualizarse la palabra ESTADISTICA. Se hace modificando adecuadamente el fichero Site.Master.
4. Diseña la nueva página Web, de forma que se visualice el resultado de tu consulta.
En la página Web diseñada, deberá aparecer un título para la consulta, el enunciado final de la SELECT y debajo la tabla con el resultado de la consulta programada. A esta tabla se le deberá dar algún formato para que quede “más elegante” en la página web. Como MÍNIMO hay que poner: colores, cambiar alguna cabecera para que no coincida con el nombre del campo en la tabla de la Base de Datos, y añadir paginación.

3. PREPARAR LA ENTREGA INDIVIDUAL Y ENTREGAR CUANDO ESTÉ ACABADO

Crea una carpeta llamada *Apellido_Nombre_INDIV* e incluye en ella: (1) La base de datos y (2) La carpeta con la aplicación Web que contiene tu consulta en un GridView (la aplicación ha de estar programada con Visual Studio Community).

Cuando esté todo listo, comprime la carpeta y entrégala a través del link “*ENTREGA PROYECTO: Práctica P11 SELECT individual*” del egela de BD dentro del plazo establecido dependiendo de tu subgrupo de prácticas.

Importante: Si la entrega ocupa más espacio que el permitido en eGela (20 MB), borra las carpetas backup y packages de la aplicación Web. Comprímela con extensión .7z (que comprime más).

PRÁCTICA P12

PROYECTO VIDEOCLUB (4):

Programación de las páginas Web del socio y del administrador

(Final de la Tarea 1 – En grupo o individual)

ACTIVIDADES PREVIAS Y OBJETIVOS

Para realizar esta práctica se necesitan:

- La BD en ACCESS del videoclub de tu grupo.
- La aplicación Web del grupo, con la página *Catálogo PELÍCULAS* implementada.

Al terminar esta práctica se debe saber:

- Diseñar y utilizar los controles de servidor ASP.NET
- Codificar y utilizar instrucciones SQL, transacciones y datos de los controles en VB.NET.

<p>NOTA: El material necesario para la realización de esta práctica está en los apuntes (en fotocopias) y en el egela de BD (material adicional, links y videos).</p>
--

1. IMPLEMENTAR LAS PÁGINAS WEB DEL SOCIO Y DEL ADMINISTRADOR

La tarea es completar la funcionalidad de la aplicación Web del enunciado del proyecto.

MI APLICACIÓN ASP.NET

Pantalla de bienvenida **administrador!** [[Cerrar sesión](#)]

[Página principal](#) | [Acerca de](#) | **Funciones ADMINISTRADOR** | [Funciones SOCIO](#) | [Catálogo PELICULAS](#)

FUNCIONES ADMINISTRADOR

Realice todas las funciones que quiera . Rellene los datos oportunos y pulse el botón.

Cambiar estado de un socio

UsuarioLogin del socio a cambiar de estado

[Cambiar estado](#)

<Visualizar aquí los datos del socio>

[Mostrar datos socio](#)

Dar de alta una película

Código de la película a dar de alta

[Dar de alta película](#)

<Poner aquí el resto de datos a pedir>

Dar de baja una película

Código de la película a dar de baja

[Dar de baja esta película](#)

<Visualizar aquí los datos de la peli>

[Mostrar datos película](#)

[Volver a la Página Principal](#)

MI APLICACIÓN ASP.NET

Pantalla de bienvenida **administrador!** [[Cerrar sesión](#)]

[Página principal](#) | [Acerca de](#) | [Funciones ADMINISTRADOR](#) | **Funciones SOCIO** | [Catálogo PELICULAS](#)

FUNCIONES SOCIO

Realice todas las funciones que quiera . Rellene los datos oportunos y pulse el botón.

Modificar mis datos personales

Nombre y apellidos

Dirección

[Modificar datos](#)

Aumentar mi crédito

Introduzca aquí la cantidad en euros

[Aumentar crédito](#)

Alquilar película

Seleccione de la lista la película a alquilar

<insertar aquí DropDownList>

[Alquilar](#)

Devolver película

Seleccione de la lista la película a devolver

<insertar aquí DropDownList con los nombres de las pelis>

[Devolver](#)

[Volver a la Página Principal](#)

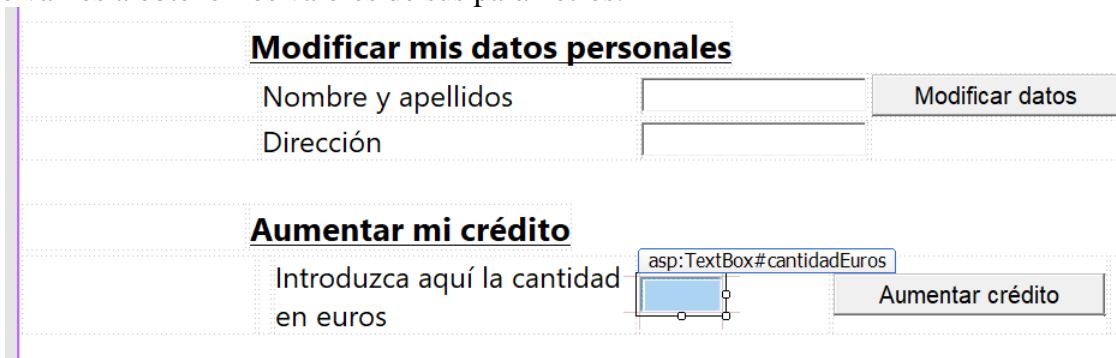
Se deben codificar las funcionalidades del administrador del video club y los socios, es decir, los botones de las pantallas y lo que sea oportuno en los mensajes en rojo (indicados entre <>) que aparecen en las figuras anteriores. Se puede ver el funcionamiento a conseguir en los vídeos *VideoClub_FUNCIONES_ADM.avi* y *VideoClub_FUNCIONES_SOCIO.avi* colgados en el egela de BD.

2. EJEMPLO DE CODIFICACIÓN DE UN BOTÓN

En el siguiente ejemplo se explican los pasos a dar para codificar un botón, particularizado para el botón con el texto “Aumentar crédito”. Todos los botones contienen alguna instrucción SQL, que seguirá la sintaxis de la base de datos utilizada, en nuestro caso el SQL de MS ACCESS.

INSTRUCCIONES SQL Y VALORES DE SUS PARÁMETROS

Lo primero es pensar qué instrucciones necesita ese botón (como cuando hacemos cualquier otro programa, es decir, variables y estructuras de control if, while, etc.), centrándonos en dónde irán las instrucciones SQL y de dónde vamos a obtener los valores de sus parámetros.



Para el caso del botón del ejemplo...

Cuando se haga clic en el botón “Aumentar crédito” se tendrá que añadir, al crédito del socio que ha iniciado sesión, el valor que haya en la caja de texto (TextBox con ID cantidadEuros). Existe una sola instrucción SQL a realizar, que es:

UPDATE SOCIO

SET CREDITO=CREDITO+ valor caja de texto

WHERE usuarioLogin= usuario que ha iniciado la sesión

donde “Valor caja de texto” es un parámetro y “usuario que ha iniciado la sesión” es otro parámetro.

El orden de los parámetros coincide con el orden en que aparecen en la instrucción SQL. Así, en este ejemplo, “Valor caja de texto” es el primer parámetro y “usuario que ha iniciado la sesión” es el segundo parámetro.

El valor de la caja de Texto lo obtendremos de *cantidadEuros.Text* que es un string.

El usuario que ha iniciado la sesión lo obtendremos de la variable *usuario*, que es una variable global de la clase en la que se guarda precisamente eso.

CODIFICACIÓN DEL BOTÓN

Para codificar un botón, se hace doble clic en el botón y sale la subrutina en la que habrá que poner el código (en la figura de nuestro ejemplo son las líneas 66, 67 y 68). En la línea 67 pondremos las instrucciones VB.NET que sean necesarias (detalladas más abajo).

```
FormularioFuncionesSOCIO.aspx.vb* x FormularioFuncionesSOCIO.aspx*
ApliWebVideoClub Aumentar Click
55 'PASO 4. CERRAR LA CONEXIÓN Y LIBERAR MEMORIA
56 conexion.Close()
57 conexion.Dispose()
58 End If
59 End Sub
60
61
62 0 referencias
63 Protected Sub VolverPrincipal_Click(ByVal sender As Object, ByVal e As EventArgs) Handles VolverPrinc:
64     Response.Redirect("default.aspx")
65 End Sub
66
67 0 referencias
68 Protected Sub Aumentar_Click(sender As Object, e As EventArgs) Handles Aumentar.Click
69 End Sub
End Class
```

Nos tenemos que fijar que esté el Imports de la línea 1 (para poder trabajar con una base de datos en MS ACCESS). Y también *que estén declaradas como variables globales* de la clase, para ser accesibles a nuestras instrucciones SQL:

- La variable *usuario* (línea 5): que contiene el nombre del usuario que ha iniciado la sesión.
- La variable *cadenaConexion* (línea 7): que indica la base de datos que se utiliza en nuestras instrucciones SQL.

```

1 Imports System.Data.OleDb 'Para las conexiones tipo OleDb -- ACCESS
2
3 Public Class FormularioFuncionesSOCIO
4     Inherits System.Web.UI.Page
5     'Asigna a Usuario el LoginName actual pasado a minúsculas (para las comparaciones)
6     Dim usuario As String = StrConv(System.Web.HttpContext.Current.User.Identity.Name, VbStrConv.Lo
7     'Indicamos la cadena de conexion (tipo OLEDB)
8     Dim cadenaConexion As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\TEMP\VideoClub

```

INSTRUCCIONES VB.NET PARA EL BOTÓN

A continuación, se codifica el botón incluyendo la programación de las instrucciones SQL necesarias. En nuestro ejemplo, el código para el botón es solamente la instrucción SQL. Hay más explicaciones en las **páginas de 13 a 23 del TEMA 5 de teoría**.

Los PASOS 1 y 4 solo se realizan una vez por botón, aunque se codifiquen varias instrucciones SQL en el botón.

Solo cambian los PASOS 2 y 3. Estos dos pasos habrá que ponerlos una vez por cada instrucción SQL que haya que codificar en el botón. A la hora de poner el PASO 3, existen 3 casos posibles y habrá que ver cuál de ellos es.

Si tenemos dudas sobre instrucciones que aparecen en el código: Buscar en Google poniendo VB.NET antes. Por ejemplo “VB.NET CDb1”. Lo mismo si queremos buscar algo que no está.

'PASO 0. DECLARAR LAS VARIABLES NECESARIAS para las instrucciones de BD

```

Dim conexion As OleDb.OleDbConnection
Dim instruccionSQL As String
Dim dbComm As OleDbCommand

```

'PASO 1. CREAR UNA CONEXION CON LA BASE DE DATOS Y ABRIRLA

```

conexion = New OleDb.OleDbConnection(cadenaConexion)
conexion.Open()

```

'PASO 2. PREPARAR LA INSTRUCCION SQL A EJECUTAR

'Ponemos la instrucción SQL en un string, con un ? en cada sitio en el que vaya un parámetro

```

instruccionSQL = "UPDATE SOCIO SET CREDITO=CREDITO + ? WHERE usuarioLogin=?"

```

'La convertimos en un comando para poder añadirle los valores de los parámetros

```

dbComm = New OleDbCommand(instruccionSQL, conexion)

```

'Añadimos los parámetros y especificamos sus valores

'en el mismo orden en que aparecen en la instrucción SQL

'Valor correspondiente a la primera ? de instruccionSQL

```

dbComm.Parameters.Add("param1", OleDbType.Double)
dbComm.Parameters("param1").Value = CDb1(cantidadEuros.Text)

```

'Valor correspondiente a la segunda ? de instruccionSQL

```

dbComm.Parameters.Add("param2", OleDbType.VarChar)
dbComm.Parameters("param2").Value = usuario

```

'PASO 3. EJECUTAR LA INSTRUCCION SQL (hay 3 casos distintos, se elige el adecuado)

```

dbComm.ExecuteNonQuery() 'Ejecuta cualquier instrucción que no sea SELECT
'Sacamos un mensaje por pantalla (opcional)
MsgBox("Se ha aumentado el crédito del socio")

```

'PASO 4. CERRAR LA CONEXIÓN CON LA BASE DE DATOS Y LIBERAR MEMORIA

```

conexion.Close()
conexion.Dispose()

```

PROBAR EL BOTÓN

Ejecutamos la aplicación desde la página maestra Site.Master y probamos el botón, para verificar que funciona correctamente.

3. CODIFICACIÓN DE DropDownList y DataList

Se hacen de la misma manera que los GridView (ver el ejemplo en la práctica P10).

Al teclear la SELECT, si la SELECT tiene parámetros se pone un ? en el lugar que va cada uno, igual que en el ejemplo anterior del botón, y luego se le indican los valores de los parámetros por orden, en la siguiente pantalla que saldrá.

Se pueden usar DataList para visualizar los datos del socio y los datos de la película en FUNCIONES ADMINISTRADOR. En este caso, los parámetros de la SELECT están en cajas de texto de la misma página web que el DataList, por lo que son parámetros de control.

Existe, en el código ya hecho, una variable de sesión llamada usuarioLogin (ver página 20 del TEMA 5 de TEORÍA). Esta variable es útil para hacer los DropDownList.

4. ELEMENTOS A CODIFICAR Y ORDEN

Se seguirá este orden:

- 1- Botón “Modificar Datos” (muy parecido a “Aumentar crédito”)
- 2- DropDownList en la caja de texto que está al lado del botón “Cambiar estado”
- 3- Botón “Cambiar estado”. ¡Cuidado con el valor que damos al parámetro!
- 4- Botón “Mostrar datos socio”. Debe hacer visible el Datalist con los datos del socio seleccionado en la caja de texto.
- 5- Botón “Dar de alta película”. Crear antes elementos para el resto de datos que hay que pedir, además del código.
- 6- DropDownList en la caja de texto que está al lado del botón “Dar de baja esta película”
- 7- Botón “Dar de baja esta película”
- 8- Botón “Mostrar datos película”. Debe hacer visible el Datalist con los datos de la película seleccionada en la caja de texto
- 9- DropDownList en la caja de texto que está al lado del botón “Alquilar”
- 10- DropDownList en la caja de texto que está al lado del botón “Devolver”

En principio, los botones que constan de más de una instrucción SQL y queremos que se hagan todas o no se haga ninguna, se codifican utilizando Transacciones. Es lo que pasa con el botón de Alquilar y el de Devolver, hay varias instrucciones SQL y no vale que se hagan unas sí y otras no (por ejemplo si se va la electricidad). Para ver qué son las transacciones y cómo se usan: **tema 5 de teoría, páginas 24, 27 y de la 31 a la 34.**

- 11- Botón “Alquilar”. Se deben incluir todas sus instrucciones SQL en una única transacción.
- 12- Botón “Devolver”. Lo mismo que para Alquilar.

Si formas parte de un grupo, esta tarea se tiene que realizar **DE MANERA CONJUNTA** entre los dos componentes del grupo. Si no formas parte de un grupo, no es necesario que implementes las funciones del administrador. En ambos casos se debe presentar lo realizado antes de entregar.

2. PREPARAR LA ENTREGA EN GRUPO Y PRESENTAR ANTES DE ENTREGAR

Cada grupo debe crear una carpeta llamada *Apellido1_Y_Apellido2_PROYECTO* (donde Apellido1 y Apellido2 son los primeros apellidos de los componentes del grupo en orden alfabético), e incluir en ella:

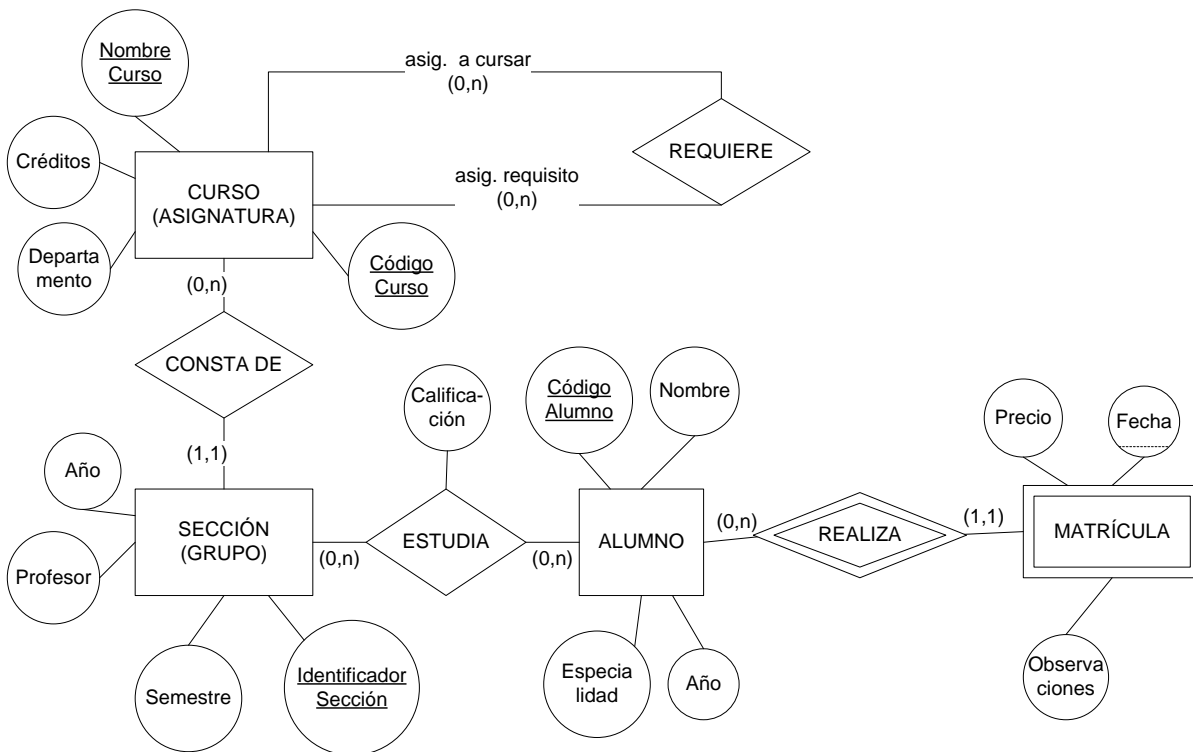
1. El nombre de los componentes del grupo, más el esquema E/R y el paso a tablas para la base de datos del videoclub del grupo (en un .pdf ó .docx).
2. La base de datos del grupo *Videoclub_Apellido1_Y_Apellido2.accdb* (compatible con MS ACCESS 2007) con datos suficientes para que se pueda probar lo programado.
3. La carpeta solución de la aplicación Web que contiene las páginas Web programadas: Funciones Administrador, Funciones Socio y Catálogo películas. Esta aplicación ha de estar programada con Visual Studio 2013 ó con Visual Studio Community.

Cuando esté todo listo, presenta el trabajo al profesor. A continuación, comprime la carpeta y entrégala a través del link “*ENTREGA PROYECTO WEB*” del egela de BD dentro del plazo establecido.

Si has hecho este trabajo *sin un grupo* (individualmente) debes seguir los mismos pasos para entregar.

Importante: Si la entrega ocupa más espacio que el permitido en egela (20 MB), borra las carpetas backup y packages de la aplicación Web. Comprímela con extensión .7z (que comprime más).

BD UNIVERSIDAD



MATRÍCULA(Código Alumno, Precio, Fecha, Observaciones)

ALUMNO(Nombre, Código Alumno, Año, Especialidad)

INFORME_CALIFICACIONES(Código Alumno, Identificador Sección, Calificación)

SECCIÓN(Identificador Sección, Código Curso, Semestre, Año, Profesor)

CURSO(Nombre Curso, Código Curso, Créditos, Departamento)

REQUISITO(Código Curso, Código requisito)

ALUMNO

Nombre	Código alumno	Año	Especialidad
Smith	17	1	CS
Brown	8	2	CS
Juan	22	1	CS
Pepe	23	3	IT

INFORME_CALIFICACIONES

Código alumno	Identificador sección	Calificación
8	85	A
8	92	A
8	102	B
8	135	A
17	112	B
17	119	C
22	112	A
22	135	A
23	112	D

CURSO

Nombre Curso	Código Curso	Créditos	Departamento
Introd. a la computación	CS1310	4	LSI
Estructuras de datos	CS3320	4	LSI
Matemáticas Discretas	MATE2410	3	MATE
Bases de Datos	CS3380	6	LSI
Administración de BD	CS3390	6	LSI
Diseño de BD	CS3400	6	LSI

REQUISITO

Código curso	Código requisito
CS3380	CS3320
CS3380	MATE2410
CS3320	CS1310
CS3390	CS3380
CS3400	CS3380

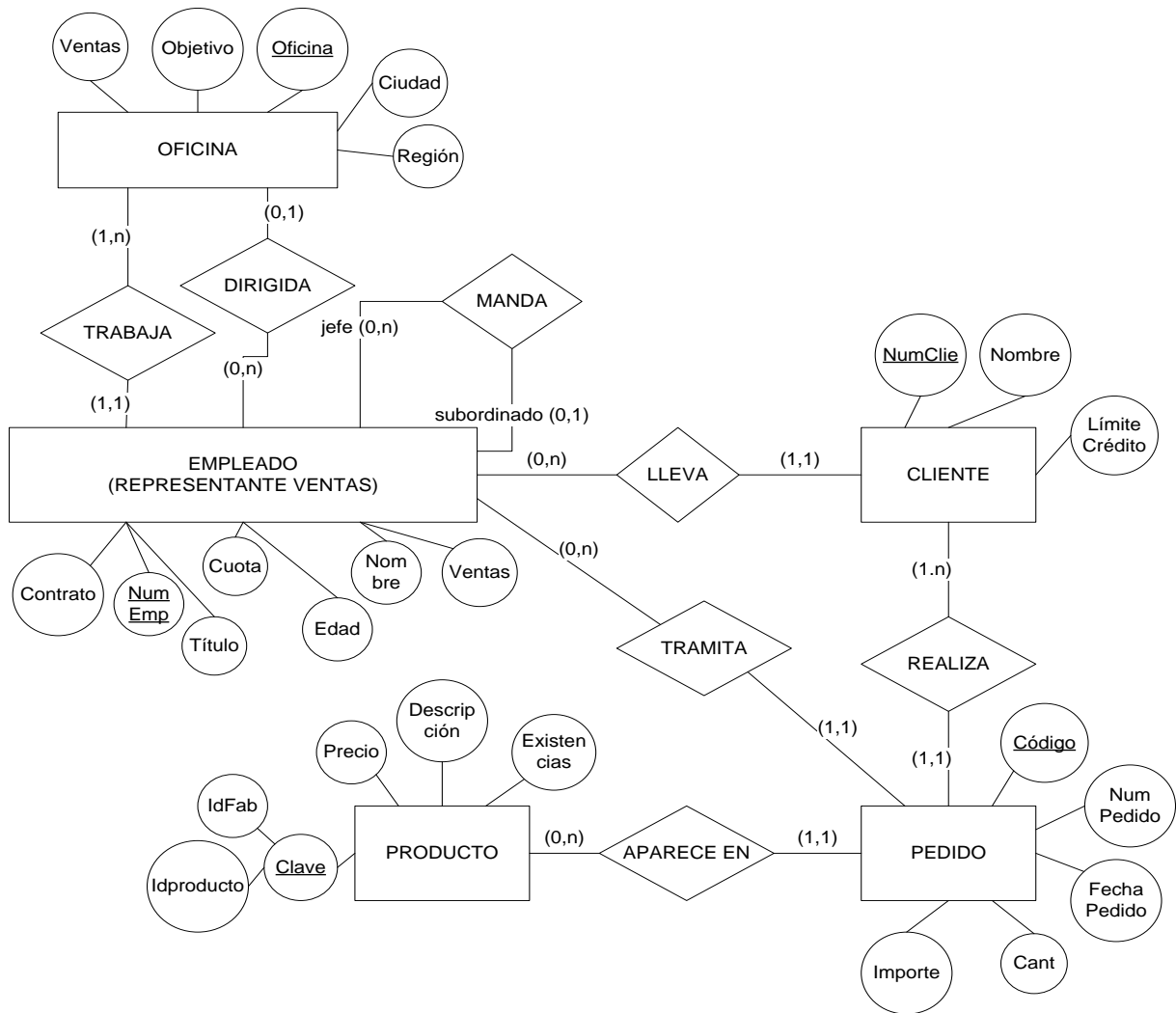
SECCIÓN

Identificador sección	Código Curso	Semestre	Año	Profesor
85	MATE2410	Otoño	1998	King
92	CS1310	Otoño	1998	Anderson
102	CS3320	Primavera	1999	Knuth
112	MATE2410	Otoño	1999	Chang
119	CS1310	Otoño	1999	Anderson
135	CS3380	Otoño	1999	Stone

MATRICULA

Código alumno	Precio	Fecha_Hora	Observaciones
8	1000	1/10/2002 10:00:00	Falta fotocopia del carné de identidad.
17	800	15/9/2002 12:15:00	

BD REPRES_VENTAS



OFICINAS(oficina, ciudad, región, dir, objetivo, ventas)

EMPLEADOS(numemp, nombre, edad, oficina, título, contrato, jefe, cuota, ventas)

CLIENTES(numclie, nombre, repclie, limitecrédito)

PEDIDOS(código, numpedido, fechapedido, clie, rep, fab, producto, cant, importe)

PRODUCTOS(idfab, idproducto, descripción, precio, existencias)

ESTADO DE LA BASE DE DATOS REPRES_VENTAS

EMPLEADOS

num-emp	nombre	edad	oficina	titulo	contrato	jefe	cuota	ventas
101	Antonio Viguer	45	12	representante	20-oct-86	104	3.000,00 euros	3.050,00 euros
102	Alvaro Jaumes	48	21	representante	10-dic-86	108	3.500,00 euros	4.740,00 euros
103	Juan Rovira	29	12	representante	1-mar-87	104	2.750,00 euros	2.860,00 euros
104	José González	33	12	dir ventas	19-may-87	106	2.000,00 euros	1.430,00 euros
105	Vicente Pantalla	37	13	representante	12-feb-88	104	3.500,00 euros	3.680,00 euros
106	Luis Antonio	52	11	dir general	14-jun-88		2.750,00 euros	2.990,00 euros
107	Jorge Gutierrez	49	22	representante	14-nov-88	108	3.000,00 euros	1.860,00 euros
108	Ana Bustamante	62	21	dir ventas	12-ene-89	106	3.500,00 euros	3.610,00 euros
109	María Sunta	31	11	representante	12-oct-99	106	3.000,00 euros	3.920,00 euros
110	Juan Víctor	41		representante	13-ene-90	104		760,00 euros

CLIENTES

numclie	nombre	repclie	limitecredito
2101	Luis García Antón	106	650,00 euros
2102	Alvaro Rodriguez	101	650,00 euros
2103	Jaime Llorens	105	500,00 euros
2105	Antonio Canales	101	650,00 euros
2106	Juan Suarez	102	650,00 euros
2107	Julián López	110	350,00 euros
2108	Julia Antequera	109	550,00 euros
2109	Alberto Juanes	103	250,00 euros
2111	Cristóbal García	103	500,00 euros
2112	María Silva	108	500,00 euros
2113	Silvia Marón	104	200,00 euros
2114	Cristina Bulini	102	200,00 euros
2115	Vicente Martínez	101	200,00 euros
2117	Carlos Tena	106	350,00 euros
2118	Junípero Alvarez	108	600,00 euros
2119	Salomón Bueno	109	250,00 euros
2120	Juan Malo	102	500,00 euros
2121	Vicente Ríos	103	450,00 euros
2122	José Marchante	105	300,00 euros
2123	José Libros	102	400,00 euros
2124	Juan Bolto	107	400,00 euros

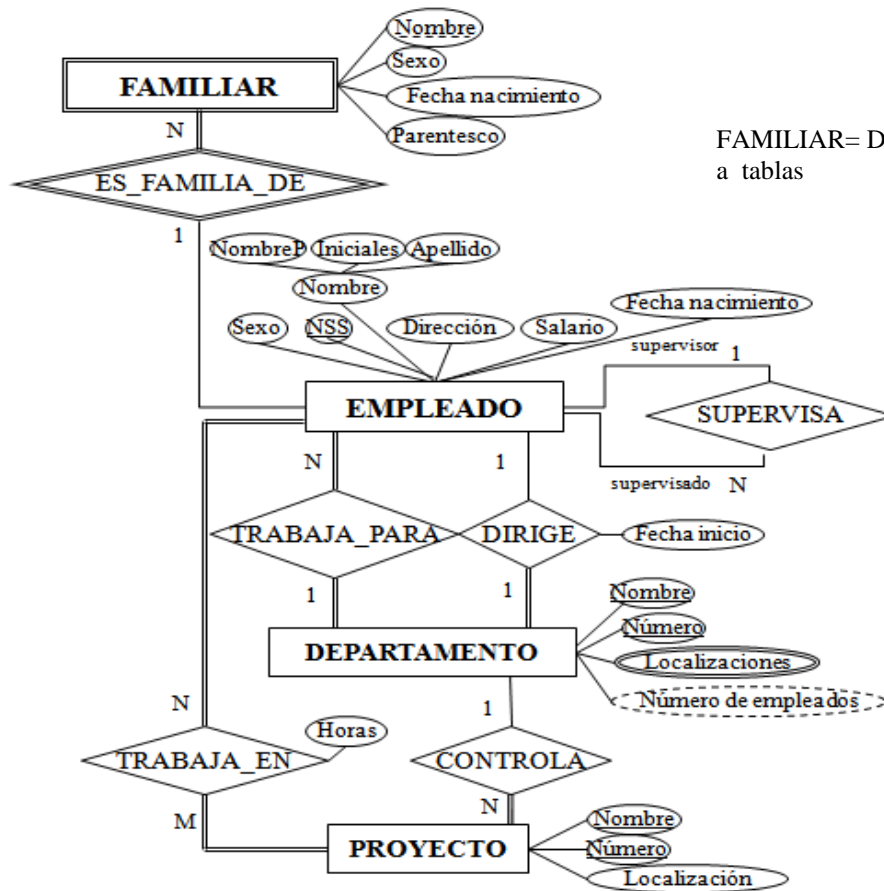
OFICINAS

oficina	ciudad	region	dir	objetivo	ventas
11	Valencia	este	106	6.760,00 euros	6.990,00 euros
12	Alicante	este	104	8.000,00 euros	7.350,00 euros
13	Castellón	este	105	3.500,00 euros	3.680,00 euros
21	Badajoz	oeste	108	7.250,00 euros	8.360,00 euros
22	A Coruña	oeste	101	1.000,00 euros	1.000,00 euros
23	Madrid	centro	108	2.500,00 euros	1.500,00 euros
26	Pamplona	norte			
28	Valencia	este		9.000,00 euros	

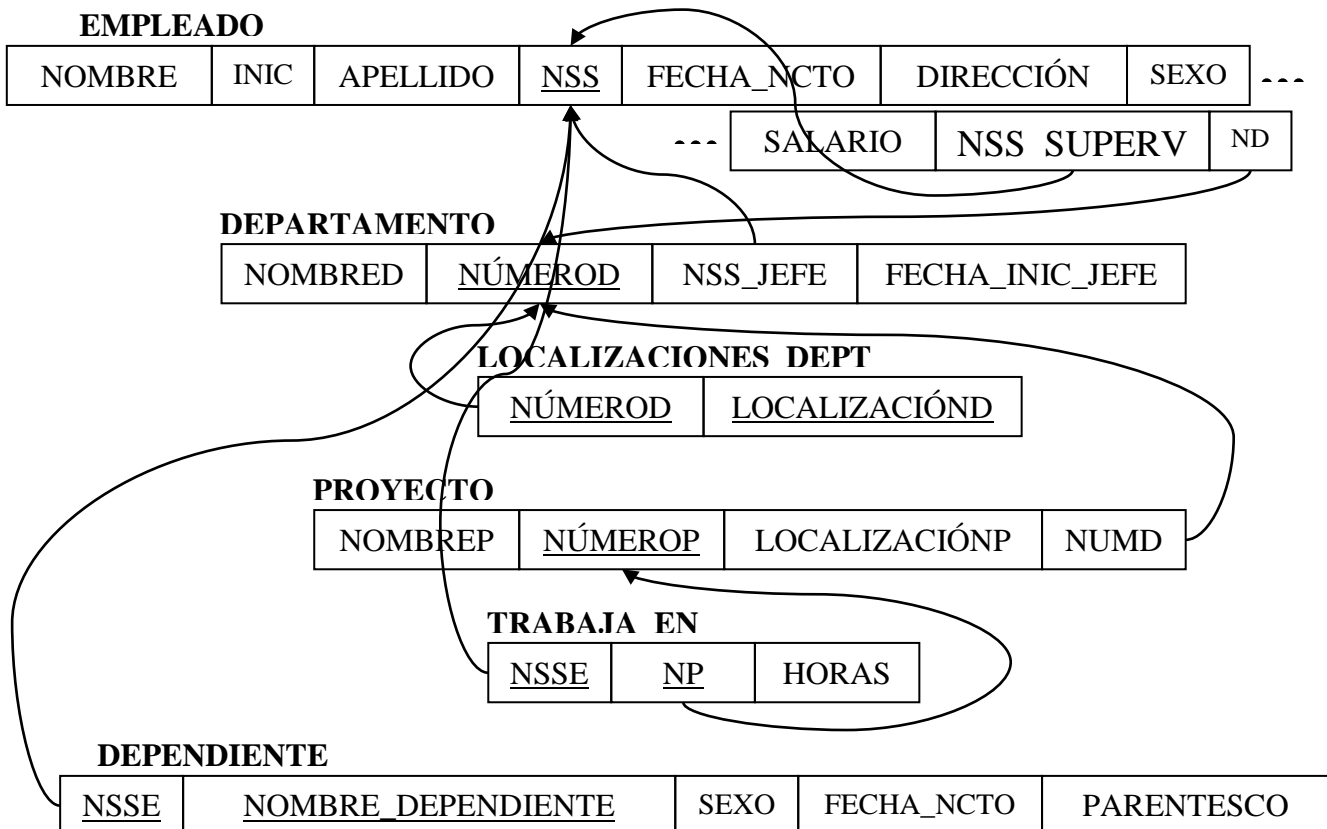
PRODUCTOS	idfab	idproducto	descripcion	precio	existencias
	aci	41001	arandela	0,58 euros	277
	aci	41002	bisagra	0,80 euros	167
	aci	41003	art t3	1,12 euros	207
	aci	41004	art t4	1,23 euros	139
	aci	4100x	junta	0,26 euros	37
	aci	4100y	extractor	28,88 euros	25
	aci	4100z	mont	26,25 euros	28
	bic	41003	manivela	6,52 euros	3
	bic	41089	rodamiento	2,25 euros	78
	bic	41672	plato	1,80 euros	0
	fea	112	cubo	1,48 euros	115
	fea	114	cubo	2,43 euros	15
	imnn	773c	reostato	9,75 euros	28
	imnn	775c	reostato 2	14,25 euros	5
	imnn	779c	reostato 3	18,75 euros	0
	imnn	887h	caja clavos	0,54 euros	223
	imnn	887p	perno	0,25 euros	24
	imnn	887x	manivela	4,75 euros	32
	qsa	xk47	red	3,55 euros	38
	qsa	xk48	red	1,34 euros	203
	qssa	xk48a	red	1,17 euros	37
	rei	2a44g	pas	3,50 euros	14
	rei	2a44l	bomba l	45,00 euros	12
	rei	2a44r	bomba r	45,00 euros	12
	rei	2a45c	junta	0,79 euros	210

PEDIDOS	codigo	numpedido	fechapedido	clie	rep	fab	producto	cant	importe
	1	1100036	2-ene-02	2107	110	aci	4100z	9	225,00
	2	1100036	2-ene-02	2117	105	rei	2a44l	7	315,00
	3	112963	10-may-02	2103	105	aci	41004	28	32,76
	4	112968	11-ene-95	2102	101	aci	41004	34	39,78
	5	112975	11-feb-02	2111	103	rei	2a44g	6	21,00
	6	112979	12-oct-94	2114	108	aci	4100z	6	150,00
	7	112963	10-may-02	2103	105	aci	41004	6	7,02
	8	112987	1-ene-02	2103	105	aci	4100y	11	275,00
	9	112989	10-dic-02	2101	106	fea	114	6	14,58
	10	112992	15-abr-95	2118	108	aci	41002	10	7,60
	11	112993	10-mar-02	2106	102	rei	2a45c	24	18,96
	12	112997	4-abr-02	2124	107	bic	41003	1	6,52
	13	113003	5-feb-02	2108	109	imnn	779c	3	56,25
	14	113007	1-ene-02	2112	108	imnn	773c	3	29,25
	15	113012	5-may-02	2111	105	aci	41003	35	37,45
	16	113013	6-ago-02	2118	108	bic	41003	1	6,52
	17	113024	4-jul-02	2114	108	qsa	xk47	20	71,00
	18	113027	5-feb-02	2103	105	aci	41002	54	41,04
	19	113034	5-nov-02	2107	110	rei	2a45c	8	6,32
	20	113042	1-ene-02	2113	101	rei	2a44r	5	225,00
	21	113045	2-jul-02	2112	108	rei	2a44r	10	450,00
	22	113048	2-feb-02	2120	102	imnn	779c	2	37,50
	23	113049	4-abr-02	2118	108	qsa	xk47	2	7,75
	24	113051	6-jul-02	2118	106	qsa	xk47	4	14,20
	25	113055	1-abr-95	2108	101	aci	4100x	6	1,50
	26	113067	1-nov-02	2111	103	aci	4100x	24	6,00
	27	113058	4-jul-94	2108	109	fea	112	10	14,80
	28	113062	4-jul-02	2124	107	bic	41003	10	24,30
	29	113065	3-jun-02	2106	102	qsa	xk47	6	21,30

BD EMPRESA



FAMILIAR= DEPENDIENTE al pasarlo a tablas



EMPLEADO

NOMBRE	INIC	APELLIDO	NSS	FECHA_NCTO	DIRECCIÓN
John	B	Smith	123456789	1965-01-09	Fresnos 731, Houston, TX
Franklin	T	Wong	333445555	1955-12-08	Valle 638, Houston, TX
Alicia	J	Zelaya	999887777	1968-07-19	Castillo 3321, Sucre, TX
Jennifer	S	Wallace	987654321	1941-06-20	Bravo 291, Bellaire, TX
Ramesh	K	Narayan	666884444	1962-09-15	Espiga 875, Heras, TX
Joyce	A	English	453453453	1972-07-31	Rosas 5631, Houston, TX
Ahmad	V	Jabbar	987987987	1969-03-29	Dalias 980, Houston, TX
Jaime	E	Borg	888665555	1937-11-10	Sorgo 450, Houston, TX

...

TRABAJA_EN

NSSE	NP	HORAS
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	nulo

...

SEXO	SALARIO	NSS_SUPERV	ND
H	30.000	333445555	5
H	40.000	888665555	5
M	25.000	987654321	4
M	43.000	888665555	4
H	38.000	333445555	5
M	25.000	333445555	5
H	25.000	987654321	4
H	55.000	nulo	1

LOCALIZACIONES_DEPT

NÚMEROD	LOCALIZACIÓN
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

DEPARTAMENTO

NOMBRE	NÚMEROD	NSS_JEFE	FECHA_INIC_JEFE
Investigación	5	333445555	1988-05-22
Administración	4	987654321	1995-01-01
Dirección	1	888665555	1981-06-19

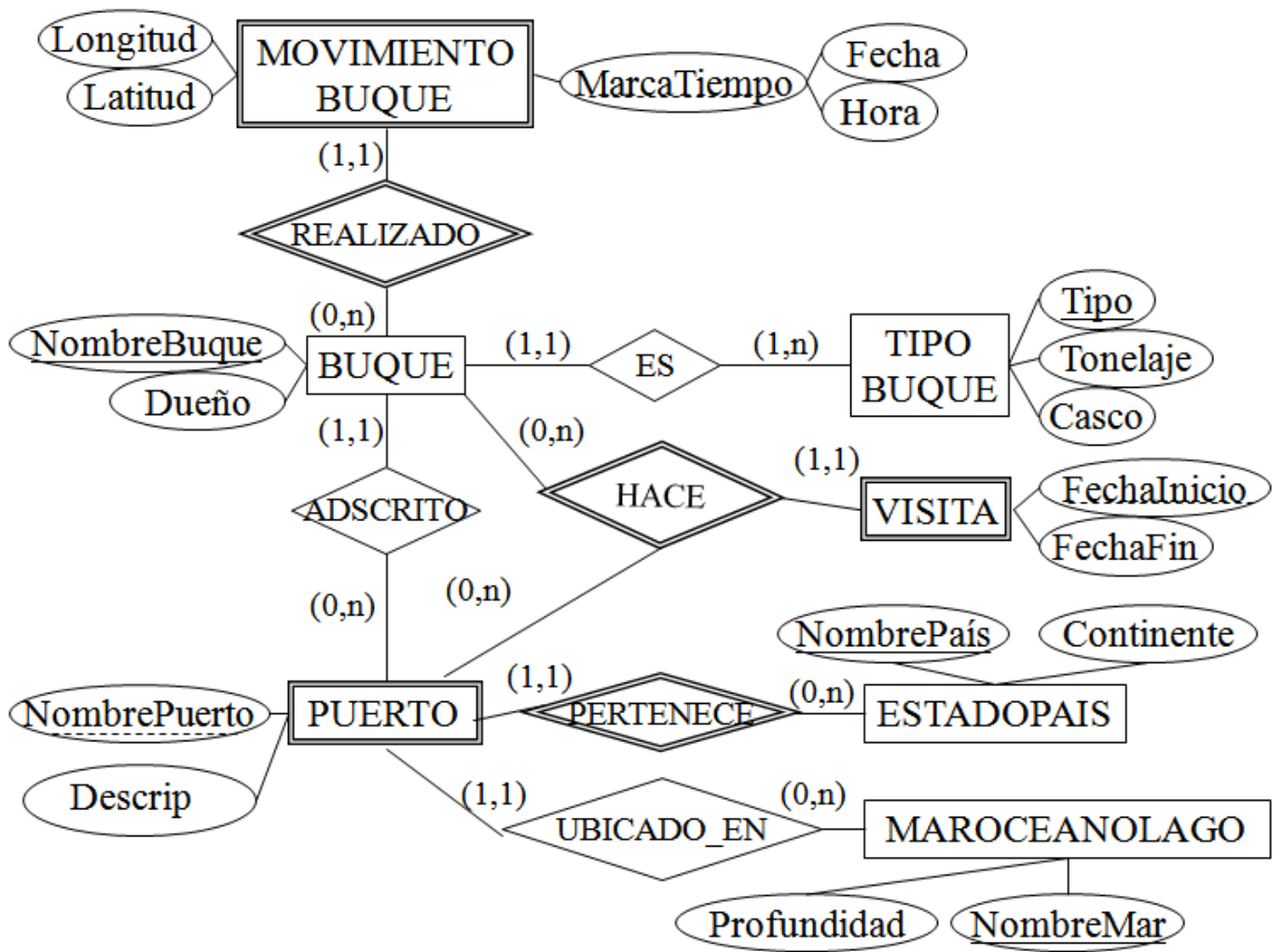
PROYECTO

NOMBREP	NÚMEROP	LOCALIZACIÓNP	NUMD
ProductoX	1	Bellaire	5
ProductoY	2	Sugarland	5
ProductoZ	3	Houston	5
Automatización	10	Stafford	4
Reorganización	20	Houston	1
Nuevos beneficios	30	Stafford	4

DEPENDIENTE

NSSE	NOMBRE_DEPENDIENTE	SEXO	FECHA_NCTO	PARENTESCO
333445555	Alicia	M	1986-04-05	HIJA
333445555	Theodore	H	1983-10-25	HIJO
333445555	Joy	M	1958-05-03	ESPOSA
987654321	Abner	H	1942-02-28	ESPOSO
123456789	Michael	H	1988-01-04	HIJO
123456789	Alicia	M	1988-12-31	HIJA
123456789	Elizabeth	M	1967-05-05	ESPOSA

BD BUQUES



MovimientoBuque

<u>NombreBuque</u>	<u>Fecha</u>	<u>Hora</u>	Longitud	Latitud
--------------------	--------------	-------------	----------	---------

TipoBuque

<u>Tipo</u>	Tonelaje	Casco
-------------	----------	-------

Buque

<u>NombreBuque</u>	<u>Dueño</u>	Tipo	<u>NombrePuerto</u>	<u>NombrePaís</u>
--------------------	--------------	------	---------------------	-------------------

EstadoPaís

<u>NombrePaís</u>	Continente
-------------------	------------

Puerto

<u>NombrePuerto</u>	<u>NombrePaís</u>	NombreMar	Descrip
---------------------	-------------------	-----------	---------

MarOcéanoLago

<u>NombreMar</u>	Profundidad
------------------	-------------

Visita

<u>NombreBuque</u>	<u>NombrePuerto</u>	<u>NombrePaís</u>	<u>FechaInicio</u>	<u>FechaFin</u>
--------------------	---------------------	-------------------	--------------------	-----------------

TIPOBUQUE

TIPO	TONELAJE	CASCO
10	3500	FibraVidrio
20	50000	Acero
30	500	Madera
40	1	Plástico

BUQUE

NOMBREBUQUE	DUENO	TIPO	NOMBREPUERTO	NOMBREPAIS
Bilbao	Onassis	10	LaRochelle	France
AndraMari	Kennedy	20	Detroit	USA
Fletan	Kirk	20	Vancouver	Canadá
Osakaka	OsakaSan	30	Osaka	Japón
Fly	Smith	40	Oslo	Noruega

VISITA

NOMBREBUQUE	NOMBREPUERTO	NOMBREPAIS	FECHAINICIO	FECHAFIN
AndraMari	Santander	España	22/11/00 0:00:00	22/11/00 0:00:00
AndraMari	Vigo	España	23/11/00 0:00:00	24/11/00 0:00:00
AndraMari	Cadiz	España	25/11/00 0:00:00	25/11/00 0:00:00
Bilbao	Santander	España	1/1/01 0:00:00	2/1/01 0:00:00
Bilbao	Vigo	España	18/1/01 0:00:00	18/1/01 0:00:00
Bilbao	Cadiz	España	5/1/01 0:00:00	9/1/01 0:00:00
AndraMari	Cadiz	España	26/11/00 0:00:00	30/11/00 0:00:00
Fletan	Santander	España	1/5/00 0:00:00	9/5/00 0:00:00
Fletan	Cadiz	España	9/5/00 0:00:00	18/5/00 0:00:00
Osakaka	Santander	España	26/3/00 0:00:00	27/3/00 0:00:00
Osakaka	Vigo	España	28/3/00 0:00:00	29/3/00 0:00:00
Osakaka	Cadiz	España	2/4/00 0:00:00	7/4/00 0:00:00

ESTADOPAIS

NOMBREPAIS	CONTINENTE
France	Europa Occidental
USA	América
México	América
UK	Europa Occidental
Rumania	Europa del Este
Japón	Asia
Noruega	Europa del Norte
Canadá	América
España	Europa del Sur

MAROCEANOLAGO

NOMBREMAR	PROFUNDIDAD
Atlántico	6000
Cantábrico	3300
Egeo	1500
Índico	4700
Mediterrán	2000
Negro	1300
Pacífico	5200
Rojo	1100
Eire	1000
Superior	1200
Aral	2000
Norte	2100

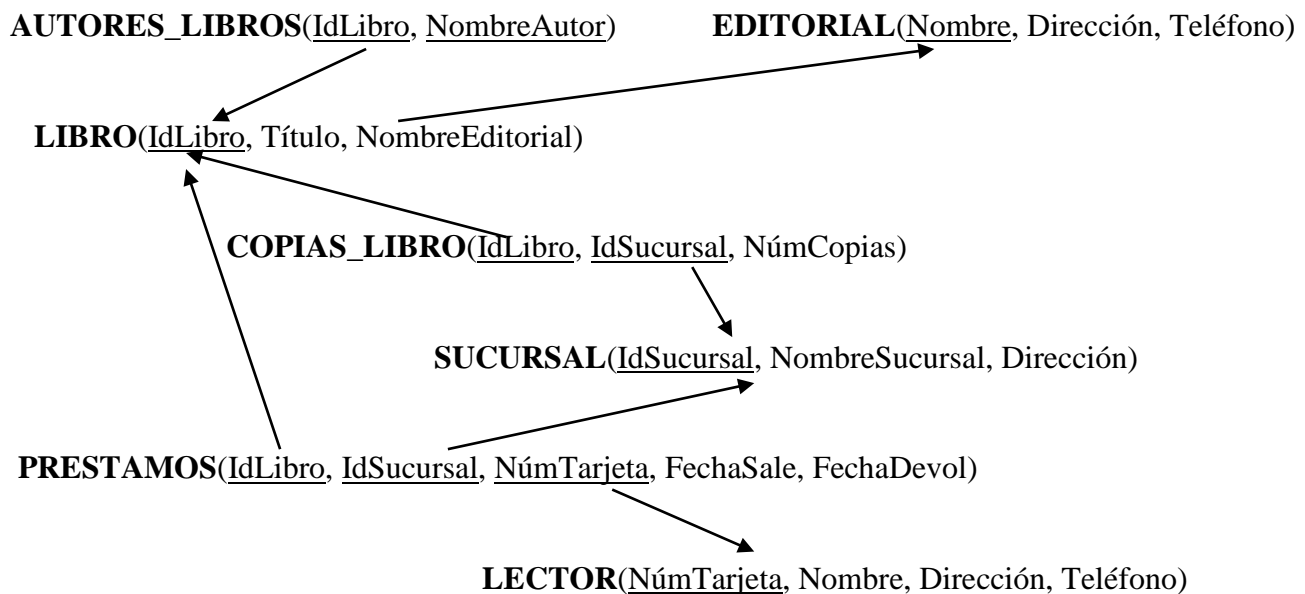
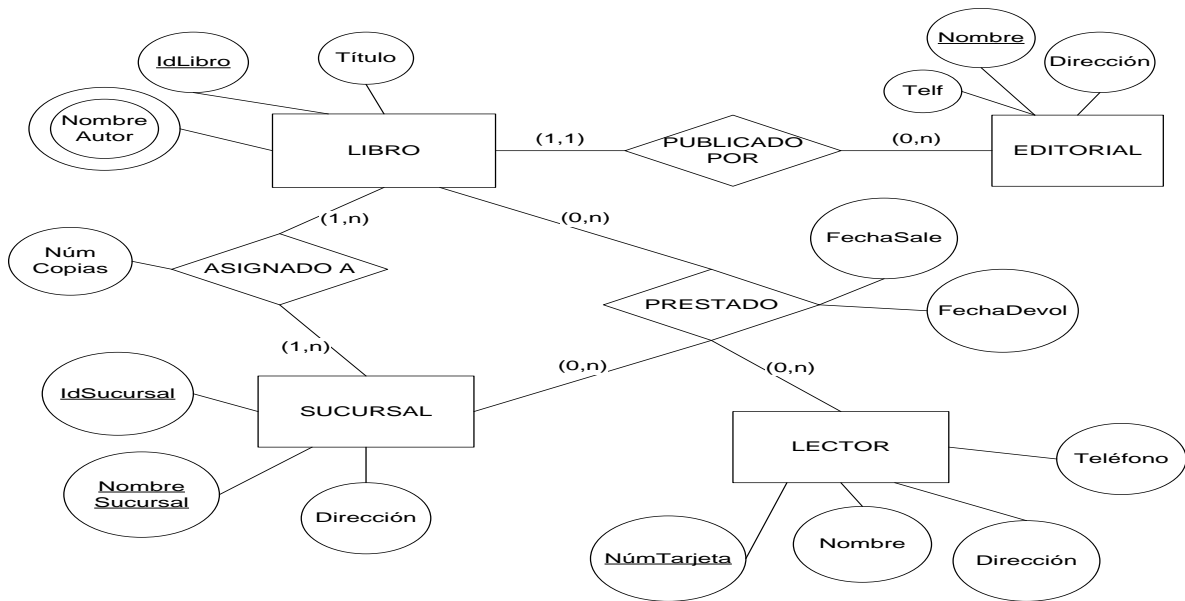
MOVIMIENTOBUQUE

NOMBREBUQUE	FECHA HORA	LONGITUD	LATITUD
Osakaka	1/1/01 0:00:00	100	150
Osakaka	2/1/01 0:00:00	125	110
AndraMari	3/1/00 0:00:00	100	100

PUERTO

NOMBREPUERTO	NOMBREPAIS	NOMBREMAR	DESCRIP
LaRochelle	France	Atlántico	Cap. 2000 barcos
Osaka	Japón	Pacífico	Cap. 300 barcos
Vancouver	Canadá	Pacífico	Rocas en el fondo
Detroit	USA	Eire	
Oslo	Noruega	Norte	Hielos
Santander	España	Cantábrico	
Vigo	España	Atlántico	Chapapote
Cadiz	España	Atlántico	Fuertes vientos

BD BIBLIOTECA



AUTORES_LIBROS

IDLIBRO	NOMBREAUTOR
100	ELMASRI
100	NAVATHE
101	STEPHEN KING
102	STEPHEN KING
103	STEPHEN KING
104	STEPHEN KING
105	STEPHEN KING
106	STEPHEN KING
107	STEPHEN KING
108	DAN BROWN
109	DAN BROWN

EDITORIAL

NOMBRE	DIRECCION	TELEFONO
ADDISON WESLEY	NULL	NULL
ANAYA	NULL	NULL
CIRCULO DE LECTORES	NULL	NULL
DEBOLSILLO	NULL	NULL
PRENTICE-HALL	NULL	NULL

LIBRO

IDLIBRO	TITULO	NOMBREEDITORIAL
100	Sistemas de Bases de Datos	ADDISON WESLEY
101	El Resplandor	DEBOLSILLO
102	It	DEBOLSILLO
103	Carrie	DEBOLSILLO
104	La Torre Oscura I	CIRCULO DE LECTORES
105	La Torre Oscura II	CIRCULO DE LECTORES
106	La Torre Oscura III	CIRCULO DE LECTORES
107	La Torre Oscura IV	DEBOLSILLO
108	ElCodigo Da Vinci	CIRCULO DE LECTORES
109	Angeles y Demonios	CIRCULO DE LECTORES
110	La Tribu Perdida	NULL

SUCURSAL

IDSUCURSAL	NOMBRESUCURSAL	DIRECCION
1	EUITI-BI	La Casilla 3 - Bilbao
2	CENTRAL	Casco Viejo - Bilbao
3	SHARPSTOWN	Picadilly Circus - London

COPIAS_LIBRO**LECTOR**

IDLIBRO	IDSUCURSAL	NUMCOPIAS	NUMTARJETA	NOMBRE	DIRECCION	TELEFONO
100	1	15	10000	PEPE	NULL	NULL
101	2	15	10001	ANA	NULL	NULL
101	3	120	10002	JON	NULL	NULL
102	2	7				
102	3	40				
103	2	1				
103	3	14				
104	2	1				
105	2	1				
106	3	1				
107	3	1				
108	2	5				
109	2	17				
110	1	3				
110	2	38				
110	3	33				

PRESTAMOS (en la BD que usaremos algunas de estas fechas cambian)

IDLIBRO	IDSUCURSAL	NUMTARJETA	FECHASALE	FECHADEVOL
101	2	10001	2020-02-14 11:30:27	NULL
102	2	10001	2019-12-01 00:00:00	2019-12-08
102	2	10001	2020-01-15 11:30:27	2020-01-16
103	2	10001	2018-12-01 00:00:00	2018-12-08
103	2	10001	2020-01-25 11:30:27	2020-02-04
103	2	10001	2020-02-07 11:30:27	NULL
103	3	10000	2019-10-01 00:00:00	2019-10-03
103	3	10000	2020-01-17 11:30:27	2020-01-17
104	2	10001	2018-12-01 00:00:00	2018-12-08
105	2	10001	2018-12-01 00:00:00	2018-12-08
106	3	10001	2018-05-01 00:00:00	2020-01-11
108	2	10001	2018-05-01 00:00:00	2018-05-11
108	2	10001	2020-02-03 00:00:00	2020-02-11