

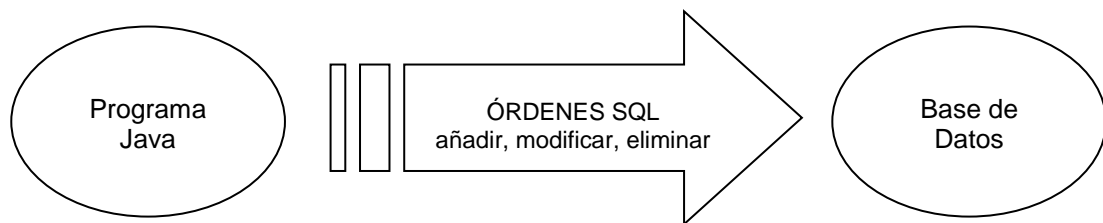
EJERCICIO GUIADO. BASES DE DATOS. SQL

Acciones en SQL

En la hoja guiada anterior se ha visto como se pueden crear instrucciones usando el lenguaje SQL para consultar una base de datos, y, de esta forma, extraer información de ella.

Pero el lenguaje SQL no se queda simplemente ahí, sino que proporciona instrucciones que pueden realizar acciones sobre la base de datos. Estas acciones son (entre otras):

- Añadir nuevos registros a una tabla.
- Eliminar registros de una tabla.
- Modificar datos de una tabla.



En esta hoja guiada veremos como ejecutar instrucciones SQL que permitan añadir, eliminar o modificar registros de las tablas de la base de datos.

Alta de registros en SQL

Se empezará estudiando como añadir nuevos registros en una base de datos usando el lenguaje SQL.

Código base en SQL para añadir nuevos registros

Para añadir un registro en una tabla se usa la siguiente sintaxis:

```
INSERT INTO tabla  
VALUES (valor1, valor2, valor3, ..., valor n);
```

En la cláusula INSERT INTO se indica la tabla en la que se quiere introducir una nueva fila (registro), y en la cláusula VALUES se indican los valores de la fila que se quiere insertar. Estos valores tienen que estar indicados en el orden en el que están definidos los campos en la tabla correspondiente.

Por ejemplo:

```
INSERT INTO trabajadores  
VALUES ('30.234.234-A', 'María', 'López', 1250.45, #01/02/2006# '4455-RSD');
```

En esta instrucción se está introduciendo un nuevo trabajador en la tabla trabajadores. Concretamente se está introduciendo un trabajador con las siguientes características:

DNI: 30.234.234-A
Nombre: María
Apellidos: López
Sueldo: 1250,45
Fecha de entrada: 02/01/2006
Matrícula: 4455-RSD

Los valores están indicados en el mismo orden en el que están los campos en la tabla: DNI, Nombre, Apellidos, Sueldo, Fecha, Matrícula.

Si se quiere introducir un nuevo registro, pero indicando solo los valores de algunos campos, se puede usar entonces esta otra sintaxis para la instrucción INSERT INTO:

```
INSERT INTO tabla  
(campo a, campo b, campo c)  
VALUES (valor del campo a, valor del campo b, valor del campo c);
```

En este caso solo se introducen los valores correspondientes a tres campos, el resto de los campos del registro se quedarán vacíos.

Por ejemplo:

```
INSERT INTO trabajadores  
(DNI, Apellidos, Sueldo)  
VALUES ('30.234.234-A', 'López', 1250.45);
```

En este caso introducimos a un trabajador de apellido López, con DNI 30.234.234-A que cobra 1250,45 euros. Tanto la matrícula de su coche como su nombre y la fecha quedan en blanco.

Formato de los valores.

Recuerda que debes seguir ciertas reglas para introducir valores:

Las cadenas se escriben entre comillas simples. Por ejemplo: 'López'

Se usa el punto decimal en los números reales: Por ejemplo: 1250.45

Las fechas se indicarán entre almohadillas # y hay que indicar primero el mes, luego el día y finalmente el año. Por ejemplo: #12-20-2007# es 20 de diciembre de 2007.

En el caso de querer especificar explícitamente que un campo esté vacío, se puede indicar el valor NULL.

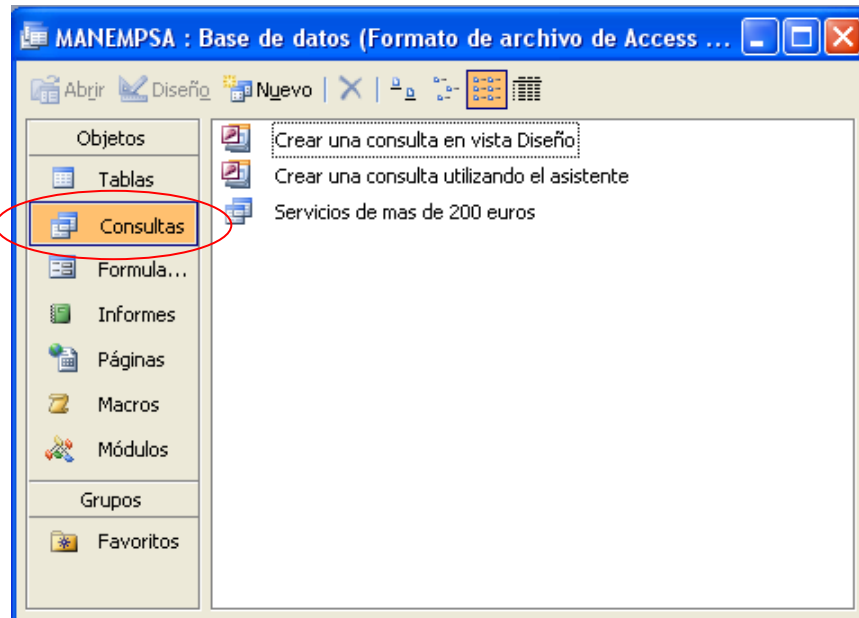
Ejemplo:

```
INSERT INTO trabajadores  
(DNI, Apellidos, Sueldo)  
VALUES ('30.234.234-A', NULL, 1250.45);
```

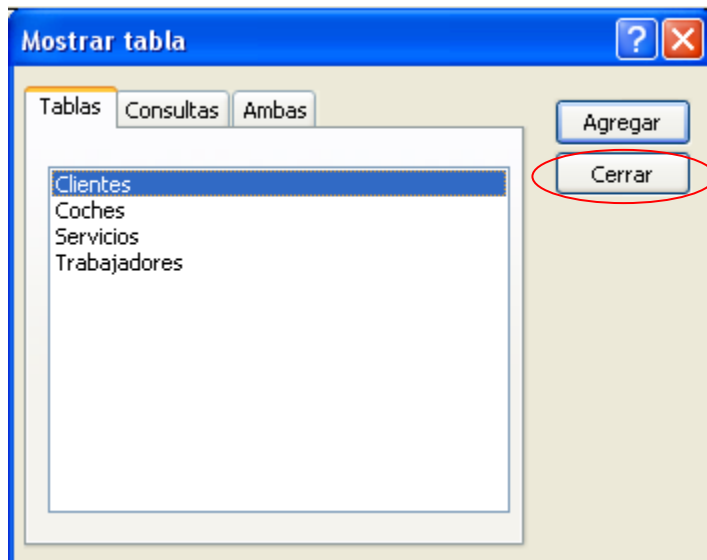
Los apellidos del trabajador están vacíos en este ejemplo.

EJERCICIO GUIADO Nº 1

1. Entra en la base de datos MANEMPSA de la carpeta Mis Documentos.
2. Crearemos una consulta SQL para añadir registros a la tabla trabajadores. Para ello, accede a la zona de consultas.



3. Crea una consulta en vista Diseño y no agregues ninguna tabla. Simplemente pulsa el botón Cerrar.

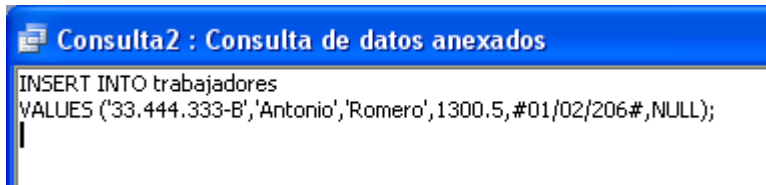


4. Pulsa el botón SQL que se encuentra en la parte superior izquierda de la ventana para crear una consulta SQL.



5. Aparecerá entonces el editor de consultas SQL. En él, escribe la siguiente instrucción SQL:

```
INSERT INTO trabajadores  
VALUES ('33.444.333-B', 'Antonio', 'Romero', 1300.5, #01/02/2006#, NULL);
```



Verás que lo que se pretende es introducir un nuevo trabajador. Este trabajador tendrá los siguientes datos:

DNI: 33.444.333-B

Nombre: Antonio

Apellidos: Romero

Sueldo 1300,5

Fecha de entrada: 2-1-2006

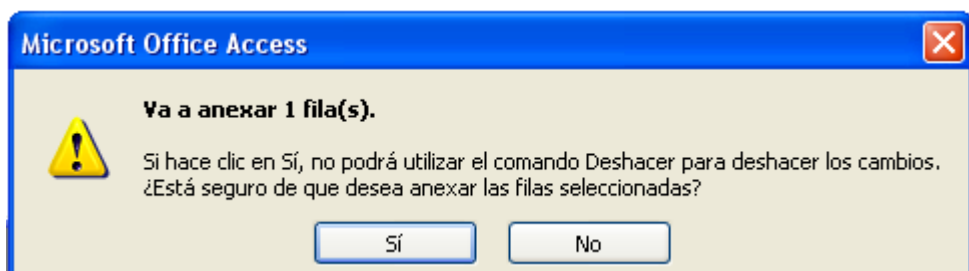
(Observa como en la cláusula VALUES se indica la fecha en formato mes-día-año)

Matrícula de su coche: No se indica (NULL)

6. Este tipo de consulta es de acción. Esto significa que no debe mostrar nada, simplemente ejecuta una acción sobre la base de datos (en este caso, añadir un registro nuevo en la tabla trabajadores) Para ejecutar esta consulta de acción activa el botón *Ejecutar*.



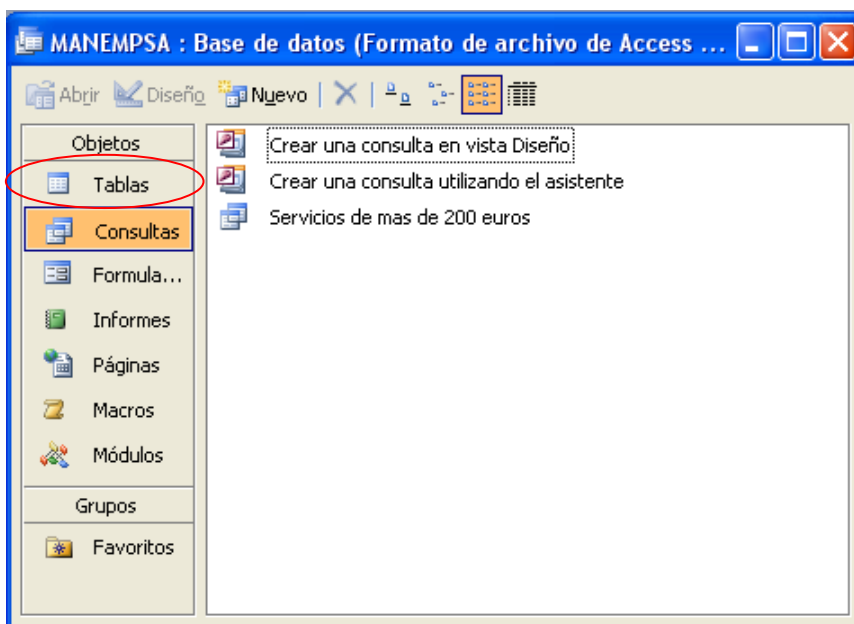
7. Access avisará de que vas a añadir un nuevo registro, y te pide confirmación. Acepta la acción:



8. Se acaba de ejecutar la consulta. Para comprobar el resultado tendremos que ver el contenido de la tabla *Trabajadores* y observaremos que hay un nuevo trabajador añadido. Primero cierra la consulta, pero no la guardes.



9. Accede a la zona de *tablas* de la base de datos.



10. Abre la tabla *trabajadores*. Observa el nuevo trabajador que se ha añadido:

Trabajadores : Tabla						
	DNI	Nombre	Apellidos	Sueldo	Fecha	Matricula
	12.321.567-B	Juan	Pérez	1120	04/05/2002	4433-ABB
	21.123.123-A	Ana	Ruiz	1200	02/03/2002	3322-ASR
	22.333.444-C	Francisco	López	1000	01/06/2006	1144-BBB
	33.444.333-B	Antonio	Romero	1300,5	02/01/2006	

11. Cierra la base de datos.

Modificación de registros en SQL

Es posible usar el lenguaje SQL para modificar los datos de una tabla. Se puede incluso modificar los datos de aquellos registros que cumplan una condición en concreto.

Código base en SQL para modificar registros

Para modificar los datos de los registros de una tabla se usará el siguiente código general:

```
UPDATE tabla a modificar  
SET campo1 = nuevovalor1, campo2 = nuevovalor2, ..., campon = nuevovalor n  
WHERE condición;
```

En la cláusula UPDATE se indica la tabla cuyos registros se van a modificar.

La cláusula SET permite indicar los cambios que se realizarán. Se debe indicar el campo que se va a cambiar y el nuevo valor que se introducirá en el campo. Como puede observar, se pueden indicar varios campos a modificar.

La cláusula WHERE permite indicar una condición. Esto hará que solo se cambien los valores de aquellos registros que cumplan la condición. La cláusula WHERE es opcional, y si no se indicara se cambiarían todos los registros de la tabla.

Ejemplo:

```
UPDATE trabajadores  
SET sueldo = 1200, matricula='3355-AAA'  
WHERE fecha < #01/01/2004#;
```

En este ejemplo se les asigna un sueldo de 1200 euros y el coche con matrícula 3355-AAA a todos aquellos trabajadores que hayan entrado en la empresa antes del 1-1-2004.

Ejemplo:

```
UPDATE trabajadores  
SET sueldo = 1300;
```

En este ejemplo, se les asigna a todos los trabajadores un sueldo de 1300 euros ya que no se ha indicado cláusula WHERE.

Si se quisiera hacer un cambio puntual a un registro en concreto, tendremos que hacer uso de su campo clave para indicar la condición. Recuerda que el campo clave es el que identifica de forma única a un registro de la tabla.

Por ejemplo:

```
UPDATE trabajadores  
SET sueldo = 1300  
WHERE DNI='33.444.333-A';
```

En este ejemplo se estoy asignando un sueldo de 1300 al trabajador con DNI 33.444.333-A, y a ningún otro (ya que se supone que no habrá otro con dicho DNI)

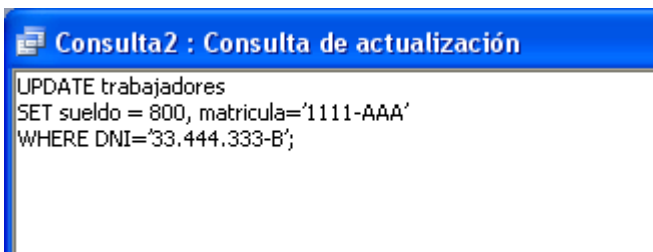
EJERCICIO GUIADO Nº2

1. Entra en la base de datos MANEMPISA.
2. Accede a la zona de consultas y crea una consulta en vista diseño.
3. No agregues ninguna tabla y pulsa directamente el botón *Cerrar*.
4. Pulsa el botón SQL para entrar en la zona de edición de consulta SQL.



5. Escribe la siguiente instrucción SQL:

```
UPDATE trabajadores  
SET sueldo = 800, matricula='1111-AAA'  
WHERE DNI='33.444.333-B';
```



Como puedes observar, en esta instrucción SQL vamos a asignarle un sueldo de 800 euros y el coche con matricula 1111-AAA al trabajador con DNI 33.444.333-B, que es precisamente el trabajador que añadimos en el ejercicio guiado anterior.

6. Ejecuta esta consulta de acción pulsando el botón *Ejecutar*.
7. Access avisará de que va a actualizar una fila en la tabla *trabajadores* y pedirá confirmación para hacerlo. Acepte la confirmación.
8. El resultado de esta instrucción SQL de acción es que se habrán modificado los datos del trabajador con DNI 33.444.333-B. Para comprobarlo accederemos a la tabla *trabajadores* y observaremos su contenido.
9. Cierre la consulta que está realizando (no hace falta que la guarde)
10. Acceda a la zona de tablas de la base de datos y entre en la tabla *trabajadores*, observe su contenido. El trabajador con DNI 33.444.333-B habrá cambiado:

Trabajadores : Tabla						
	DNI	Nombre	Apellidos	Sueldo	Fecha	Matricula
	12.321.567-B	Juan	Pérez	1120	04/05/2002	4433-ABB
	21.123.123-A	Ana	Ruiz	1200	02/03/2002	3322-ASR
	22.333.444-C	Francisco	López	1000	01/06/2006	1144-BBB
	33.444.333-B	Antonio	Romero	800	02/01/2006	1111-AAA

11. Cierra la base de datos.

Eliminación de registros en SQL

Al igual que podemos añadir nuevos registros (filas) a las tablas usando SQL, también podemos usar este lenguaje para eliminar registros de las tablas.

Código base en SQL para eliminar registros

Para eliminar registros de una tabla se usará el siguiente código general:

```
DELETE FROM tabla de la que se quiere eliminar  
WHERE condición de los registros que se eliminarán;
```

En la cláusula DELETE FROM se indica la tabla de la que eliminaremos registros. En la cláusula WHERE se indica la condición que deben cumplir los registros que eliminaremos.

Por ejemplo:

```
DELETE FROM trabajadores  
WHERE sueldo>1000;
```

En este ejemplo se están eliminando de la tabla *trabajadores* aquellos trabajadores cuyo *sueldo* sea superior a 1000.

Si se quiere eliminar un solo registro de la tabla, será necesario hacer referencia a su campo clave. Recuerda que el campo clave es el que identifica de forma única a cada registro.

Por ejemplo, si queremos eliminar un trabajador en concreto, indicaremos su DNI en la condición:

```
DELETE FROM trabajadores  
WHERE DNI='33.444.555-A';
```

Esta instrucción SQL borra al trabajador con DNI 33.444.555-A (solamente a él, ya que se supone que no habrá otro trabajador que tenga ese mismo DNI)

En la cláusula WHERE de una instrucción DELETE, las condiciones se indican tal como se vio en la hoja anterior para las instrucciones SELECT.

Si se quiere eliminar todo el contenido de una tabla, se puede usar una instrucción DELETE sin indicar ninguna condición. Por ejemplo:

```
DELETE FROM trabajadores;
```

Esta instrucción eliminaría todos los registros de la tabla *trabajadores*.

EJERCICIO GUIADO Nº 3

11. Entra en la base de datos MANEMPISA.
12. Accede a la zona de consultas y crea una consulta en vista diseño.
13. No agregues ninguna tabla y pulsa directamente el botón *Cerrar*.
14. Pulsa el botón SQL para entrar en la zona de edición de consulta SQL.



15. Escribe la siguiente instrucción SQL:

```
DELETE FROM trabajadores  
WHERE DNI = '33.444.333-B';
```

Como puedes observar, esta instrucción eliminará al trabajador con DNI 33.444.333-B, que es precisamente el trabajador que se añadió en el ejercicio guiado 1.

16. Ejecuta esta instrucción de acción activando el botón *Ejecutar*:



17. Access avisará de que va a eliminar una fila de la tabla trabajadores, y pedirá confirmación. Confirme la eliminación.
18. Para comprobar el ejemplo tendrá que acceder a la tabla *trabajadores* y comprobar que el trabajador con DNI 33.444.333-B ya no está. Para ello cierre la consulta (no hace falta que la guarde) y acceda a la zona de tablas.
19. Abra la tabla trabajadores. Observe como el trabajador ha desaparecido:

Trabajadores : Tabla						
	DNI	Nombre	Apellidos	Sueldo	Fecha	Matricula
	12.321.567-B	Juan	Pérez	1120	04/05/2002	4433-ABB
	21.123.123-A	Ana	Ruiz	1200	02/03/2002	3322-ASR
	22.333.444-C	Francisco	López	1000	01/06/2006	1144-BBB
▶				0		

20. Cierre Access.

CONCLUSIÓN

El lenguaje de consulta SQL no solo permite extraer información de una base de datos, sino que también permite realizar acciones sobre la base de datos:

- Introducir nuevos datos en la base de datos.**
- Eliminar datos de la base de datos.**
- Modificar datos de la base de datos.**

Para introducir nuevos datos en la base de datos usará la instrucción SQL llamada INSERT INTO

Para eliminar datos de la base de datos usará la instrucción SQL llamada DELETE

Para modificar datos de la base de datos usará la instrucción SQL llamada UPDATE