

12.B. SQL.

Sitio: [Formación Profesional a Distancia](#)

Curso: Programación

Libro: 12.B. SQL.

Imprimido por: Iván Jiménez Utiel

Día: domingo, 17 de mayo de 2020, 16:27

Tabla de contenidos

[1. Creación de las tablas en una base de datos.](#)

[2. Lenguaje SQL \(I\).](#)

[3. Lenguaje SQL \(II\).](#)

1. Creación de las tablas en una base de datos.

En Java podemos conectarnos y manipular bases de datos utilizando JDBC. Pero la creación en sí de la base de datos debemos hacerla con la herramienta específica para ello. Normalmente será el administrador de la base de datos, a través de las herramientas que proporcionan el sistema gestor, el que creará la base de datos. No todos los drivers JDBC soportan la creación de la base de datos mediante el lenguaje de definición de datos (**DDL**).

Normalmente, cualquier sistema gestor de bases de datos incluye asistentes gráficos para crear la base de datos con sus tablas, claves, y todo lo necesario.

También, como en el caso de MySQL, o de Oracle, y la mayoría de sistemas gestores de bases de datos, se puede crear la base de datos, desde la línea de comandos de MySQL o de Oracle, con las sentencias SQL apropiadas.

Vamos utilizar el sistema gestor de base de datos MySQL, por ser un producto gratuito y de fácil uso, además de muy potente.

Debes conocer

Puedes descargarte e instalar la herramienta gráfica que permite entre otras cosas trabajar para crear tablas, editarlas, añadir datos a las tablas, etc., con MySQL. Aquí puedes descargarlo:

[Descarga de MySQL WorkBench.](#)

Nota: De nuevo insistiremos en la conveniencia, si no quieres complicarte mucho con la creación de bases de datos, de tablas, la inserción, eliminación y modificación de registros, de instalar XAMPP o WAMP, y lanzar el servicio MySQL. Con él, podremos gestionar nuestras bases de datos a través de una [interfaz](#) gráfica y amigable. Concretamente a través de phpMyAdmin, aplicación que viene incluida con XAMPP o con WAMP.

2. Lenguaje SQL (I).

¿Cómo le pedimos al Sistema Gestor de Bases de Datos Relacional (SGBDR), en concreto en este caso, al de MySQL, que nos proporcione la información que nos interesa de la base de datos?

Se utiliza el **lenguaje SQL** (Structured Query Language) para interactuar con el SGBDR.

SQL es un lenguaje **no procedimental** en el cual se le indica al SGBDR **qué** queremos obtener y **no cómo hacerlo**. El SGBDR analiza nuestra orden y si es correcta sintácticamente la ejecuta.

El estudio de SQL nos llevaría mucho más que una unidad, y es [objeto](#) de estudio en otros módulos de este ciclo formativo. Pero como resulta imprescindible para poder continuar, haremos una mínima introducción sobre él.

Los comandos SQL se pueden dividir en dos grandes grupos:

- Los que se utilizan para **definir las estructuras de datos**, llamados comandos DDL (Data Definition Language).
- Los que se utilizan para **operar con los datos almacenados en las estructuras**, llamados DML (Data Manipulation Language).

En el siguiente enlace encontrarás algunos de los comandos SQL más utilizados:

[Sentencias SQL básicas.](#)

Autoevaluación

Con las consultas SQL hay que especificar paso a paso cómo recuperar cada dato de la base de datos.

- ☐ Verdadero.
- ☐ Falso.

3. Lenguaje SQL (II).

La primera fase del trabajo con cualquier base de datos comienza con sentencias DDL, puesto que antes de poder almacenar y recuperar información debemos definir las estructuras donde agrupar la información. Las estructuras básicas con las que trabaja SQL son las tablas.

Como hemos visto antes, una tabla es un conjunto de celdas agrupadas en filas y columnas donde se almacenan elementos de información.

Antes de llevar a cabo la creación de una tabla conviene planificar: nombre de la tabla, nombre de cada columna, tipo y tamaño de los datos almacenados en cada columna, información adicional, restricciones, etc.

Hay que tener en cuenta también ciertas restricciones en la formación de los nombres de las tablas: longitud. Normalmente, aunque dependen del sistema gestor, suele tener una longitud máxima de 30 caracteres, no puede haber nombres de tabla duplicados, deben comenzar con un carácter alfabético, permitir caracteres alfanuméricos y el guión bajo '_', y normalmente no se distingue entre mayúsculas y minúsculas.

Por ejemplo para crear una tabla de departamentos podríamos hacer:

```
CREATE TABLE departa (  
  cod_dep number(3),  
  nombre varchar2(15) not null,  
  loc varchar2(10),  
  constraint dep_pk primary key (cod_dep),  
  constraint dep_loc check  
    (loc in ('Madrid', 'Barcelona', 'Murcia'))  
);
```

donde creamos la tabla con cod_dep como [clave primaria](#). Además, se añade una restricción para comprobar que cuando se esté dando de alta un [registro](#), lo que se escriba en el campo **loc** sea Madrid, Barcelona o Murcia.

Y una tabla de empleados, teniendo en cuenta el departamento en el que trabajen:

```
CREATE TABLE emp (  
  cod_emp number(3),  
  nombre varchar2(10) not null,  
  oficio varchar2(11),  
  jefe number(3),  
  fecha_alta date,  
  salario number(10),  
  comision number(10),  
  cod_dep number(3),
```

```
constraint emp_pk primary key (cod_emp),
```

```
constraint emp_fk foreign key (cod_dep) references departa(cod_dep)
```

```
on delete cascade,
```

```
constraint emp_ck check (salario > 0)
```

```
);
```

En el caso de esta tabla, se puede ver que hay una restricción para comprobar que el salario sea mayor que 0.