

12.B. SQL.

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.

