

SQL 2

Sirve para crear una tabla, como su nombre indica. Su sintaxis en SQL es:

```
CREATE TABLE <nombre_tabla> (<atributo_1>, ...<atributo_n>, <restricción_1>, ...  
<restricción_n>);
```

Tipos de datos

- **Caracteres:**
 - **Varchar(n)**: cadenas de n caracteres.
 - **Clob(n)**: cadenas de más de 4000 elementos.
 - **Varchar2**: no estándar
- **Números:**
 - **Numeric(m, n)**: no es obligatorio n. Permite m número, donde n es la cantidad de decimales y m-n la cantidad de números de la parte entera.
 - **Number**: no estándar
- **Fechas:**
 - **Date**: fecha+hora

El tipo de dato Date es un poco peculiar. Por ejemplo, para mostrar solo la fecha en [Tablas empleados](#) podemos usar:

```
select hiredate  
from emp;
```

Pero si queremos un formato determinado con los componentes tenemos que usar to_char. Por ejemplo:

```
select to_char(hiredate, 'DD/MM/YYYY HH24:MI:SS')  
from emp;
```

Pasa lo mismo a la hora de insertar una fecha. Para hacerlo solo con la fecha podemos usar una simple string.

```
insert into tabla  
values ('12/12/2012');
```

Pero para incluir el resto de componentes tenemos que usar to_date. Por ejemplo:

```
insert into tabla  
values (to_date('23:00:00', 'HH24:MI:SS'));
```

Default

Sirve para indicar el valor que toma ese atributo al crear una nueva fila si no se especifica un valor. Por defecto, si no se pone default, el valor será null. Por ejemplo:

```
create table emp_prueba(  
    empno numeric(4),  
    ename varchar(20),  
    sal numeric(7,2) default 1400  
);
```

Restricciones

Son condiciones que se usan para obligar a cumplir detalles que los datos tienen en el mundo real. Sirven para garantizar la integridad de los datos. Pueden ser de columna o de tabla y se les debe dar nombre para poder quitar la restricción con ALTER TABLE si fuera necesario. Estas son:

- **Not null:**

Es una restricción de atributo que no permite que sea nulo. Por ejemplo:

```
create table emp_prueba(  
    empno numeric(4),  
    ename varchar(20) constraint r_ename not null  
);
```

- **Primary key:**

Puede usarse tanto como restricción de atributo como de tabla. Indica la clave primaria. Por ejemplo:

```
create table emp_prueba(  
    empno numeric(4) constraint r_empno primary key  
);
```

```
create table emp_prueba(  
    diaHora date,  
    id_puesto numeric(4),
```

```
empno numeric(4),
primary key (empno, diaHora)
);
```

El resto de restricciones de tabla también tienen una sintaxis similar, por lo que a partir de ahora los ejemplos van a ser generalmente con restricciones de columna.

- **Unique:**

Puede usarse tanto como restricción de atributo como de tabla. Sirve como clave candidata, ya que no permite duplicados. Permite null, por lo que debemos acompañarlo de not null. Por ejemplo:

```
create table emp_prueba(
    empno numeric(4) constraint r_empno primary key,
    ename varchar(20) unique not null
);
```

- **Foreign key:**

Puede usarse tanto como restricción de atributo como de tabla. Sirve como clave foránea. Por ejemplo:

```
create table emp_prueba(
    empno numeric(4) constraint r_empno primary key,
    ename varchar(20) unique not null,
    deptno numeric(2) default 10 references dept(deptno)
);
//en este caso, deptno de la tabla dept debe ser primary key o unique
```

```
create table emp_prueba(
    empno numeric(4) constraint r_empno primary key,
    ename varchar(20) unique not null,
    deptno numeric(2) default 10 references dept
);
//en este caso, dept debe tener una primary key
```

Si a mayores le damos la opción ON DELETE CASCADE, podríamos borrar el atributo al que apuntan las foráneas y todas las tuplas con foráneas que apuntan a él:

```
create table emp_prueba(
    empno numeric(4) constraint r_empno primary key,
```

```
        ename varchar(20) unique not null,  
        deptno numeric(2) default 10 references dept(deptno) on delete  
cascade  
);  
//ahora podríamos hacer delete from dept where deptno=10, por ejemplo
```

Otra posibilidad es dejar el valor a null al borrar en cascada. Esto se hace con ON DELETE SET NULL:

```
create table emp_prueba(  
    empno numeric(4) constraint r_empno primary key,  
    ename varchar(20) unique not null,  
    deptno numeric(2) default 10 references dept(deptno) on delete set  
null  
);
```

- **Check:**

Puede usarse tanto como restricción de atributo como de tabla. Permite solo añadir datos que cumplan la condición como true o null. Por ejemplo:

```
create table emp_prueba(  
    empno numeric(4),  
    ename varchar(20),  
    sal numeric(7,2) default 1500,  
    constraint r_sal check (sal>=1500)  
);
```