

CONSULTAS DDL

```
CREATE TABLE tabla
  (campo1 tipo [(tamaño)]
    [DEFAULT valor]
    [restricció11] [restricció12] [...] ,
  camp2 tipo [(tamaño)]
    [DEFAULT valor]
    [restricció21] [restricció22] [...]
  [...]]
[, restricciómultiple1 [...]])
```

```
CREATE TABLE EMPLEAT3
  (dni VARCHAR (10),
  nombre VARCHAR,
  fecha_nacimiento DATE,
  sueldo NUMERIC (6,2),
  departamento Int2,
  población VARCHAR (50) DEFAULT 'Castellón',
  data_incorporacio DATE DEFAULT CURRENT_DATE)
```

Restricciones de campo único

```
[CONSTRAINT nombre] {PRIMARY KEY | UNIQUE | NOT NULL |
REFERENCES Tabla2 [(campo1)] | CHECK ( condición )}
```

- **PRIMARY KEY** : el campo será clave principal.

```
CREATE TABLE EMPLEAT3
  (dni VARCHAR (10)
    CONSTRAINT cp_emp3 PRIMARY KEY,
  nombre VARCHAR,
  fecha_nacimiento DATE,
  sueldo NUMERIC (6,2),
  departamento Int2,
  población VARCHAR ( 50) DEFAULT 'Castellón',
  data_incorporacio DATE DEFAULT CURRENT_DATE)
```

- **UNIQUE** : el campo será único, es decir, no se podrá coger dos veces el mismo valor en este campo

```
CREATE TABLE EMPLEAT3
(dni VARCHAR (10),
nombre VARCHAR
CONSTRAINT u_nom UNIQUE,
fecha_nacimiento DATE,
sueldo NUMERIC (6,2),
departamento Int2,
población VARCHAR ( 50) DEFAULT 'Castellón',
data_incorporacio DATE DEFAULT CURRENT_DATE)
```

- **NOT NULL** : el campo no podrá coger un valor nulo

```
CREATE TABLE EMPLEAT3 (
dni VARCHAR (10),
nombre VARCHAR
CONSTRAINT nn_nom NOT NULL,
fecha_nacimiento DATE,
sueldo NUMERIC (6,2),
departamento Int2,
población VARCHAR (50) DEFAULT 'Castellón',
data_incorporacio DATE DEFAULT CURRENT_DATE)
```

- **REFERENCES** : servirá para definir que este campo es una clave externa. Tendremos que especificar obligatoriamente la tabla a la que apunta, y opcionalmente podemos poner entre paréntesis el campo de la tabla al que apunta, por defecto apuntará a la clave principal

```
CREATE TABLE DEPARTAMENTO
(num_d Int2
CONSTRAINT cp_dep PRIMARY KEY,
nom_d VARCHAR (50),
director VARCHAR (10),
fecha DATE);
```

```
CREATE TABLE EMPLEAT3 (
dni VARCHAR (10),
nombre VARCHAR,
fecha_nacimiento DATE,
sueldo NUMERIC (6,2),
departamento Int2
CONSTRAINT ce_emp3_dep
REFERENCES DEPARTAMENTO,
población VARCHAR (50) DEFAULT 'Castellón',
data_incorporacio DATE DEFAULT CURRENT_DATE);
```

Como ya se vio, hay 3 formas de actuar cuando se borra o se modifica una fila de la tabla principal que tiene asociadas filas en la tabla relacionada por medio de la clave externa.

- x **NO ACTION** : no se dejará borrar o modificar de la tabla principal si tiene alguna fila relacionada. Es la opción por defecto.
- x **CASCADE** : borrarán (o modificarán) en cascada los registros relacionados de la tabla donde está la clave externa. Se especificará con ON DELETE CASCADE o ON UPDATE CASCADE

```
CREATE TABLE EMPLEAT3
(dni VARCHAR (10),
nombre VARCHAR,
fecha_nacimiento DATE,
sueldo NUMERIC (6,2),
departamento Int2 CONS TRRAINT ce_emp3_dep
REFERENCES DEPARTAMENTO
ON DELETE CASCADE
ON UPDATE CASCADE,
población VARCHAR (50) DEFAULT 'Castellón',
data_incorporacio DATE DEFAULT CURRENT_DATE)
```

- x **SET NULL** : pondrá a nulo el campo que es clave externa de los registros que estén relacionados con el borrado o modificado de la tabla principal.

```
CREATE TABLE EMPLEAT3
(dni VARCHAR (10),
nombre VARCHAR,
fecha_nacimiento DATE,
sueldo NUMERIC (6,2),
departamento Int2
CONSTRAINT ce_emp3_dep
REFERENCES DEPARTAMENTO
ON DELETE SET NULL,
población VARCHAR (50) DEFAULT 'Castellón',
data_incorporacio DATE DEFAULT CURRENT_DATE)
```

- **CHECK** : hará una comprobación para validar los valores introducidos para este campo. La condición de validación debe ir entre paréntesis, y debe ser una expresión, normalmente de comparación del campo en cuestión con algún valor.

```
CREATE TABLE EMPLEAT3
(dni VARCHAR (10),
nombre VARCHAR,
fecha_nacimiento DATE,
sueldo NUMERIC (6,2)
CONSTRAINT sou_positiu
CHECK (sueldo > 0),
departamento Int2,
población VARCHAR (50) DEFAULT 'Castellón',
data_incorporacio DATE DEFAULT CURRENT_DATE)
```

Restricciones de campo múltiple

```
[CONSTRAINT nombre]
{PRIMARY KEY | UNIQUE | FOREIGN KEY | CHECK ( condición )}
(c11[, c12] [...])
[REFERENCES Tabla2 [(c21 [, C22] [...])]]
[ON DELETE {CASCADE | SET NULL}]
[ON UPDATE {CASCADE | SET NULL}]
```

- **PRIMARY KEY** : pondremos entre paréntesis el campo o campos (en este caso separados por comas) que serán clave principal.

```
CREATE TABLE FAMILIAR
(dni VARCHAR (10),
nombre VARCHAR,
data_n DATE,
parentesco VARCHAR (50),
CONSTRAINT cp_fam2 PRIMARY KEY (dni, nombre))
```

- **UNIQUE** : ahora pondremos entre paréntesis el o los campos que serán únicos (en su conjunto). PostgreSQL generará automáticamente un índice para esta combinación de campos.

```
CREATE TABLE EMPLEAT4
(dni VARCHAR (10),
apellidos VARCHAR,
nombre VARCHAR,
fecha_nacimiento DATE,
sueldo NUMERIC (6,2),
departamento Int2,
CONSTRAINT u_nom4
UNIQUE (apellidos, nombre))
```

- **NOT NULL** : No existe esta opción como restricción múltiple. Por lo tanto se debe definir siempre como restricción de campo único.
- **FOREIGN KEY** : servirá para definir que este o estos campos son una clave externa. Es la que más varía en su sintaxis, ya que tenemos que especificar tanto el o los campos de esta tabla que son clave externa, como la tabla a la que apunta.

```
CREATE TABLE EMPLEAT3
(dni VARCHAR (10),
nombre VARCHAR,
fecha_nacimiento DATE,
sueldo NUMERIC (6,2),
departamento Int2,
población VARCHAR (50) DEFAULT 'Castellón',
data_incorporacio DATE DEFAULT CURRENT_DATE,
CONSTRAINT ce_emp3_dep
FOREIGN KEY (departamento)
REFERENCES DEPARTAMENTO)
```

- **CHECK** : ahora la condición de validación podrá afectar a más de un campo

```
CREATE TABLE EMPLEAT3
(dni VARCHAR (10),
nombre VARCHAR,
fecha_nacimiento DATE,
sueldo NUMERIC (6,2),
departamento Int2,
población VARCHAR ( 50) DEFAULT 'Castellón',
data_incorporacio DATE DEFAULT CURRENT_DATE,
CONSTRAINT check_dates
CHECK (data_incorporacio > fecha_nacimiento) )
```

ALTER TABLE

- Para alterar la estructura de algún campo o restricción utilizaremos esta sintaxis:

```
ALTER TABLE tabla {ADD | DROP | ALTER}
{COLUMN campo | CONSTRAINT restricciónmúltiple}
```

- Para cambiar el nombre de un campo:

```
ALTER TABLE tabla RENAME
[COLUMN] campo TO nou_nom_camp
```

- Para cambiar el nombre de la tabla:

```
ALTER TABLE tabla RENAME TO nou_nom_taula
```

- Añadir campo o restricción

- Campo

```
ALTER TABLE EMPLEAT3
ADD COLUMN
    supervisor VARCHAR (10)
```

- Restriccion

```
ALTER TABLE EMPLEAT3
ADD CONSTRAINT ce_emp3_emp3
FOREIGN KEY (supervisor)
REFERENCES EMPLEAT3 (dni)
```

- Modificar un campo

```
ALTER TABLE EMPLEAT3
    ALTER COLUMN població
    TYPE VARCHAR (25)
```

Cambiar el tipo de datos es automático cuando los tipos son compatibles entre ellos. Si no lo son nos dará error, pero seguramente lo podremos esquivar con la cláusula USING, que nos permite poner a continuación el campo y aprovechamos para poner un operador de conversión de tipo (::) con esta sintaxis:

```
ALTER TABLE TABLA
    ALTER COLUMN c_ampo
        TYPE tipus_nou
        USING campo :: tipus_nou
```

Para cambiar el valor por defecto utilizaremos la sintaxis: ... ALTER COLUMN campo {SET | DROP} DEFAULT [expresión]

```
ALTER TABLE EMPLEAT3
    ALTER COLUMN població
    DROP DEFAULT
```

- Borrar campo o restricción

x Campo

```
ALTER TABLE EMPLEAT3
    DROP COLUMN supervisor
```

x Restriccion

```
ALTER TABLE EMPLEAT3  
    DROP CONSTRAINT ce_emp3_emp3;
```

- Renombrar un campo

```
ALTER TABLE EMPLEAT3  
    RENAME COLUMN data_incorporacio TO data_inc
```

- Renombrar la tabla

```
ALTER TABLE EMPLEAT3  
    RENAME TO EMP3
```

DROP TABLE

```
DROP TABLE tabla
```

INDICE

- Creación de índice

```
CREATE [UNIQUE] INDEX nom_índex  
ON mesa (c1 [ASC | DESC]  
    [, c 2 [ASC | DESC], ...]  
    [nula {FIRST | LAST}])
```

x **FIRST** : hará que en la ordenación los valores nulos vayan antes de cualquier otro valor. Esta es la opción por defecto si la orden es descendente además de crear el índice hace que sea clave principal. Evidentemente no debe haber una clave principal creada con anterioridad.

x **LAST** : los valores nulos estarán al final de todo, después de cualquier otro valor. Es la opción por defecto cuando la orden es ascendente.

```
CREATE INDEX i_dep  
ON EMPLEAT4 (departamento);
```

- Borrar un índice

```
DROP INDEX nom_índex ON tabla
```

VISTAS

```
CREATE VIEW nom_vista AS  
    subconsulta  
[WITH READ ONLY];
```

```
CREATE OR REPLACE VIEW ESTADISTICA AS  
SELECT nom_c,  
    count (nombre) AS num_p,  
    sum (población) AS pobl,  
    avg (altura) AS alt_mitjana  
FROM POBLACIONES  
GROUP BY nom_c  
ORDER BY nom_c;
```